Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A Modified Vision Transformer for Kurdish Cursive RTL Handwritten Text Recognition

Faraedwn M. Salih
faraedwn.mahmood@gmail.com
Koya University
Abdulbasit K. Al-talabani
Koya University

Research Article

Keywords: Deep Learning, Computer Vision, Pattern Recognition, Handwriting, Vision Transformers

Posted Date: April 6th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9225627/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License

Additional Declarations: No competing interests reported.

Abstract

The problem of handwritten text recognition (HTR) of low-resource cursive scripts is a major problem in document image analysis. This research aims to bridge the gap of unavailable annotated data and effective text recognition systems for Central Kurdish, also known as Sorani, which is a complex script consisting of 34 letters, rich ligatures, and context-dependent diacritical marks, and it is the first time this script is studied in HTR research like our approach. In this research, we present DASNUS, a new large-scale dataset of 11,475 annotated text lines from 867 writers across the Kurdistan region of Iraq. We also propose a deep learning framework for HTR, which combines ResNet-inspired convolutional encoders and a Vision Transformer model. The model is particularly suited for addressing variability, dependencies, and ligatures, which are inherent in Sorani script. The model was trained using several techniques, including span-based masking, geometric and photometric transformations, and depth regularization of the transformer using DropPath and LayerScale. The proposed model achieved a CER of 3.47%, and a WER of 17.37%, comparable to or even better than other state-of-the-art models for Arabic, Persian, and English HTR. In this research, we have established the first benchmark for Kurdish cursive handwriting. We have also demonstrated that well-regularized transformer-based models are capable of effectively recognizing complex, low-resource cursive scripts. The results of this research are expected to pave the way for future research in multilingual OCR, writer adaptation, synthetic data, and inclusive AI.

1. Introduction

In the fields of computer vision and pattern recognition, handwriting recognition remains one of the most technically challenging tasks [1]. Its applications vary from digitization of historical manuscripts to assistive technologies for people with disabilities [2]. Despite significant advances for well-resourced languages like English, Arabic, Chinese, and German, there is still a large gap in research and tools for a significant number of written languages. Central Kurdish (Sorani), also known as CKB, is one of them.

Sorani is used by 12–14 million people in Iraq and Iran [3]. It is written with a modified Arabic alphabet and has the same cursive writing direction and characters connected to each other from right to left like Arabic and Persian. However, it also contains unique characters for distinct sounds of the Kurdish language: ,ڕ ,ۆ ,ێ ,ۅ ,ۉ ,ې, and [5 ,4] ۀ. As a result, the Sorani alphabet contains 34 characters compared to 28 for Arabic and 32 for Persian. This makes handwriting recognition for Sorani significantly more difficult than for Latin and roughly Arabic scripts. Some of these difficulties include context-dependent characters with up to four different forms for some of them [6], long horizontal ligatures where two or more characters are connected to each other to form a single unit, and diacritical marks, which tend to be faint and misplaced in handwriting. Additionally, there is a lot of variation between writers in terms of pressure and stroke width and spacing (see Fig. 1).

Significant advancements in handwriting recognition have been made on languages with large annotated datasets. The advancements have evolved from Convolutional Neural Network – Recurrent Neural Network (CNN-RNN) architectures with Connectionist Temporal Classification (CTC) loss functions [7, 8] to Vision Transformer (ViT) architectures with self-attention mechanisms for connected cursive scripts [9, 10]. However, as far as we know, no publicly available annotated dataset for Central Kurdish handwriting at the line level was available before this research, and no previous research utilized a ViT-based architecture for handwriting recognition on the Sorani script at the line level.

This paper contributes to filling these research gaps in two aspects: a dataset and a model. First, we introduce a large-scale dataset for Central Kurdish handwriting. Second, we propose a modified ViT-based model for handwriting recognition on the Sorani script and also works well for other cursive scripts, achieving a Character Error Rate (CER) of 3.47% and a Word Error Rate (WER) of 17.37% on our proposed dataset, which are comparable with or even better than the state-of-the-art results for Arabic, Persian, and English handwriting recognition.

The rest of the paper is structured as follows: Section 2 reviews related works on handwriting recognition and transformer-based architectures for handwriting recognition. In Section 3, we introduce the proposed dataset for Central Kurdish handwriting recognition, named DASNUS. In Section 4, we explain the proposed model for handwriting recognition on the Sorani script, and in Section 5, we report the experimental results on the proposed dataset and some fair comparisons. In Section 6, we conclude this paper.

2. Related Work

2.1. Datasets for cursive script handwriting recognition

The development of handwriting recognition systems has largely depended on the availability of standardized datasets. For Arabic, the IFN/ENIT database [11] is one of the most widely used, consisting of 26,459 handwritten Tunisian town and village names from 411 writers. This dataset has become a benchmark for Arabic handwriting recognition, enabling CNN–BLSTM–CTC systems to achieve recognition rates above 92%. Another critical resource is the KHATT dataset [12], which contains 165,890 words, 589,924 characters, and 9,326 lines of text contributed by 1,000 different writers, with annotations at both the word and line level. The training of deep models that are resistant to handwriting variations has been facilitated by the availability of such datasets.

Curated resources have also been advantageous for Persian handwriting recognition. The dataset introduced by the authors of [13] allowed CNN–BLSTM–CTC models to obtain a 99.35% accuracy in character recognition. In the same vein, Urdu recognition has progressed with datasets of handwritten text lines, such as the 6,000-line dataset utilized in the research paper of [14], which facilitated the development of segmentation-free sequence modeling approaches.

Conversely, Kurdish continues to be underfunded. The KRDOH dataset [15] comprises 4,304 scanned pages, which amount to 93,612 words and 17,466 text lines from 1,076 authors. Although this is a significant contribution, its potential for end-to-end recognition is restricted by the absence of word-level annotation. Other Kurdish resources concentrate on isolated characters or numerals. More than 315,000 images of digits and isolated characters are included in the K-ZHMARA and K-PIT datasets [5], while the dataset in [16] contains 40,940 images of individual Kurdish letters that are evenly distributed

across the alphabet. These datasets are indispensable for character-level recognition; however, they fail to address the obstacle of continuous, cursive handwriting recognition. At the word level, the KHWD dataset [17] provided a notable improvement at the word level by obtaining approximately 400,000 handwritten word images from about 8,000 writers containing 10,000 Sorani Kurdish vocabulary words; this new approach provides evidence supporting the significantly greater challenge in cursive recognition of handwritten words versus isolated characters—the KHWD produced a best WER of 45.68% using an enhanced version of the MobileNetV2 model.

2.2. CNN–RNN architectures

In Arabic, Persian, and Urdu, the use of hybrid architectures of CNN, BLSTM, and CTC has been highly significant in the field of Handwritten Text Recognition (HTR). Hassan et al. [14] proposed a CNN-BLSTM architecture with CTC loss for the case of Urdu, which attained 83.69% accuracy in recognizing characters without explicit segmentation. However, the system faced problems due to the high similarity between classes and the complex placement of diacritic marks, which also applies in the case of Kurdish text.

Maalej and Kherallah [18] also attained high accuracy in recognizing characters by using a combination of CNNs for feature extraction and BLSTM for sequence modeling on the IFN/ENIT dataset for Arabic. In a subsequent study, Mutawa et al. [19] improved the accuracy of this approach by using a combination of ResNet for feature extraction and BLSTM with CTC for sequence modeling, which attained a CER of 13.2% and a WER of 27.31% on the KHATT dataset. This also proves the effectiveness of complex architectures of CNN-RNN, though with high dependence on large datasets.

In the paper [13], the authors were able to prove that a CNN-BLSTM-CTC architecture can attain a high accuracy of 99.35% in the case of Persian, thereby eliminating the need for explicit segmentation in the case of Persian text. In Kurdish, however, attempts have been limited. Ahmed et al. [20] proposed a CNN model trained on 40,940 isolated characters, achieving 83% test accuracy, but the model did not extend to word- or line-level recognition. Similarly, the authors in reference [16] focused only on character-level modeling, leaving the sequential dependencies of connected text largely unaddressed.

2.3. Transformer-based approaches

Transformer-based models have also been considered in HTR in light of learning long-range dependencies in connection with recurrent connections. In the case of Arabic, the researchers in reference [21] proposed a CNN–Transformer architecture hybrid (OCFormer) based on convolutional embeddings and self-attention. Performance was high, but it was dependent on a humongous set of synthetic data of over 30 million images, which posed applicability issues in hard-resource regimes.

More recent research has investigated applying ViTs to handwriting text recognition (HTR). In [10] span feature masking was added to the boosting of contextual learning, and the architecture proposed was able to surpass CNN–BLSTM baselines on English and German data, illustrating the promise of ViTs in handwriting recognition. In another research paper [22], it was suggested substituting CNN–RNN-based architecture with full Transformer-based architecture and achieving the best results on the KHATT dataset. This, however, needed to have been done with much pretrained knowledge and computational power.

Attention-based sequence-to-sequence models have also been tried. In [23] the authors showed that they perform well on handwriting recognition but showed substantial drops in performance on dense cursive scripts where only small amounts of training data could be had. All these works point to the conclusion that while Transformers give state-of-the-art performance, they require high amounts of labeled or synthesized data — which happens to be a big concern in Kurdish, where such data still do not exist.

2.4. Comparative analysis: Arabic, Persian, Urdu vs. Kurdish

Comparison between related scripts finds both similarities and exigent challenges. In Urdu, authors of paper [14] exemplified that segmentation-free CNN–BLSTM model-based systems could process at the stroke level while being tripped up by dot placement and letter likeness, both of which also appear in Kurdish. In Arabic scripts, IFN/ENIT [11] and KHATT [12] datasets allowed CNN–BLSTM–CTC-based systems to break the 92%-threshold in terms of accuracy. The authors in [24] presented the AHDB dataset in highlighting the value of standardized corpora in the development of the field in Arabic HTR. Mezghani et al. [25] similarly presented the multi-writer corpus in the form of AHTID/MW to accentuate its value in real-world writer-independent identification. Persian-based research in reference [13] presented that with deeply learned CNN–BLSTM–CTC systems, it was possible to get close to perfect accuracy upon being trained on finely curated datasets.

Kurdish, on the other hand, poses unique challenges. While in comparison to Arabic or Persian, the script of Kurdish excludes some characters such as ڕ ,ڤ ,ڤ, and ۆ that exist in others related scripts. More seriously, no big-scale, word-level, or line-level dataset comparable to IFN/ENIT or KHATT is available. Although KRDOH [15] represents a progression, it has no word annotation, and data such as K-ZHMARA [5] and the one in reference [16] contain only isolated symbol-based data. Therefore, Kurdish significantly trails Arabic, Persian and even Urdu in available dataset richness and benchmark material.

2.5. Motivation and research gap

From the above review, several conclusions emerge:

1. CNN–BLSTM–CTC architectures [13, 14, 18, 19] have proven highly effective for cursive handwriting recognition, but their performance depends on large, well-curated datasets.
2. Transformer-based approaches outperform CNN-RNN architectures for long sequence handling but need enormous amounts of training data.
3. Arabic, Persian, and Urdu have the advantage of standardized datasets like IFN/ENIT [11], KHATT [12], and AHDB [24], while Kurdish only has character- or word-based datasets.
4. There are script-specific challenges for Central Kurdish handwriting recognition: more characters, diacritic marks, and positional variations.

This establishes a distinct research gap. Existing CNN–RNN or Transformer approaches cannot be directly applied to Kurdish due to its low-resource status and script-specific complexity. Our work addresses this by developing a ViT-based tailored for Kurdish text-line recognition, and contributing toward dataset expansion, thereby bridging a critical gap in low-resource handwriting recognition research.

3. The DASNUS Dataset

The DASNUS dataset was collected as a part of this study and to the best of our knowledge, DASNUS (□□□□□□□□ — Central Kurdish for handwriting manuscript) is the largest publicly available dataset for recognizing handwritten Central Kurdish (Sorani) text. We made the dataset open to the public[1].

3.1. Data collection

- A total of 867 writers from the four main governorates of the Kurdistan Region of Iraq volunteered to write, and there were no restrictions on the type of pen or style of writing. Table 1 show a summary of the demographic information. Each writer filled out a structured four-page form (Fig. 2). This included a demographic page, a fixed paragraph that make sure all the letters of the Sorani alphabet, digits, and some common symbols were covered, and two randomly assigned paragraphs from a pool of 2,500 distinct Kurdish writings in page 3, and a free-writing portion in page 4. Out of 1,250 forms sent out, 867 were filled out completely and sent back, which made 3,468 pages.

Table 1
Demographic distribution of DASNUS writers (N = 867).

Category	Subgroup	Count	Percentage (%)
Age	Below 15	96	11.1
	16–25	591	68.2
	26–50	156	18.0
	More than 50	24	2.8
Address	Erbil	305	35.2
	Sulaymaniyah	507	58.5
	Kirkuk	10	1.2
	Duhok	45	5.2
Education Level	Primary	33	3.8
	Secondary	113	13.0
	High-School	163	18.8
	University	558	64.4
Gender	Male	538	62.1
	Female	329	37.9
Handedness	Right-handed	765	88.2
	Left-handed	102	11.8

3.2. Scanning and preprocessing

To keep the delicate nuances of the script, forms were scanned at 600 DPI. Using Otsu's thresholding [26], morphological closing, contour detection, and skew correction through projection profile analysis [27], an OpenCV-based pipeline subsequently binarized the data and created 11,475 annotated line pictures. The process is visually presented in Fig. 3.

3.3. Annotation and quality control

Native speakers of the Sorani language manually transcribed each line image verbatim into UTF-8. Inter-annotator agreement was 99.1% (character-level) and 97.6% (word-level) in the double-annotation of 500 randomly sampled lines. All disagreements were resolved by an expert annotator, resulting in an overall transcription accuracy that surpassed 98%.

3.4. Dataset splits

A writer-disjoint strategy is employed to divide DASNUS into training, validation, and test sets (Table 2). The agglutinative morphology of Sorani Kurdish is reflected in the out-of-vocabulary (OOV) rate of approximately 47% for both evaluation splits, which explains the greater WER disparity that was observed in our experiments.

Table 2
Dataset splits and vocabulary statistics.

Split	#Writers	#Lines	#Unique Words	Vocabulary Size	OOV (%)
Train	606	8042	10702	10702	-
Validation	131	1700	3298	12258	47.18
Test	130	1733	3532	13817	47.28

4. Methodology

We have modified a handwriting recognition model that is based on ViT and was expressly designed for the recognition of Central Kurdish (Sorani) cursive text lines. This model can also be used to train on other cursive scripts. An architecture composed of three components: a CNN feature extractor inspired by ResNet, a transformer encoder, and a CTC decoder. The complete architecture is depicted in Fig. 4.

4.1. CNN feature extractor

We utilize a custom CNN front-end that is based on the residual learning framework of He et al. [28], as opposed to the conventional patch embedding of the original ViT [29]. Two factors motivate this decision: (1) pure ViTs lack the spatial inductive bias necessary for stable convergence on small datasets, as demonstrated by Li et al. [10], who demonstrated that the removal of the CNN front-end from their HTR-VT model resulted in a validation CER increase from 3.3% to 26.6% on IAM. Additionally, the Sorani script contains small diacritical marks that are essential for character disambiguation and are easily lost through aggressive downsampling.

Our feature extractor generates 128 tokens with a dimension of 768 from a 64×512 grayscale image. A comprehensive description of the architecture is provided in Fig. 5. It is distinguished from both ResNet-18 [28] and the HTR-VT variant [10] in five significant ways:

There is no initial downsampling. We replace the 7×7 strided convolution and max-pool of ResNet-18 with a 5×5 convolution at stride (1,1) and omit the initial pooling layer entirely, thereby preserving the full input resolution and retaining fine diacritical details.

Gradual downsampling that is asymmetric. Using stride (2,1), layers 1 and 2 reduce the height by half while maintaining the breadth. Layer 2 is followed by an intermediate max-pool with stride (2,1), and stride (2,2) is only applied by layer 3. The sequential character information along the width dimension, which is essential for CTC alignment, is preserved by this height-first strategy.

Block configuration that is deeper. We employ a [2, 3, 4, 3] block configuration (12 total blocks), which is more profound than both ResNet-18 [2, 2, 2, 2] and HTR-VT [2, 2, 2]. This configuration enables us to acquire the intricate positional variants and ligature patterns of cursive Arabic-script writing.

Convolutions that are dilated during the ultimate stage. We retain the fourth stage with stride (1,1) and apply dilation rate 2, rather than removing it as in HTR-VT. This expands the receptive field to capture inter-character stroke dependencies without further spatial resolution loss.

Regularization of dropouts. A CNN-level regularization is provided in our low-resource context by the application of a dropout layer ($p = 0.3$) following the final residual stage, which is absent from both He et al. [28] and HTR-VT [10].

The expansion is delayed by the channel progression (192→192→384→768), which concentrates early layers on low-level stroke and diacritical features. The final 768-channel dimension is directly proportional to the transformer embedding size, thereby eradicating any projection layer between the two components. The final 128×768 token sequence is generated through a two-stage spatial collapse, which involves average pooling and adaptive average pooling. Table 3 provides a comprehensive comparison between our architecture and the two discussed architectures.

Table 3
Comparison of CNN feature extractor architectures.

Feature	ResNet-18	HTR-VT	Ours
Input channels	3 (RGB)	1 (grayscale)	1 (grayscale)
Initial conv kernel	7×7, stride (2,2)	3×3, stride (2,1)	5×5, stride (1,1)
Initial pooling	MaxPool, stride 2	MaxPool, stride (2,1)	None
Residual stages	4	3	4
Blocks per stage	[2, 2, 2, 2]	[2, 2, 2]	[2, 3, 4, 3]
Total residual blocks	8	6	12
Channel progression	64→128→256→512	192→384→768	192→192→384→768
Dilated convolutions	No	No	Yes (layer4, d = 2)
Dropout in CNN	No	No	Yes (p = 0.3)
Intermediate pooling	No	No	Yes (after layer2)
Final spatial collapse	Global AvgPool	MaxPool (2,1)	AvgPool + AdaptiveAvgPool
Output dimension	512	768	768

4.2. Transformer encoder

The transformer encoder simulates long-range contextual dependencies throughout the token sequence. In comparison to ViT-Base [29] and HTR-VT [10] in Table 4, our configuration employs 8 encoder layers and 8 attention centers with a 768-dimensional embedding. We adhere to the Pre-LayerNorm formulation [30]:

$$\varvec{y}^{\varvec{n}} = \varvec{x}^{(\varvec{n}-1)} + \varvec{D}\varvec{r}\varvec{o}\varvec{p}\varvec{P}\varvec{a}\varvec{t}\varvec{h}\left(\varvec{L}\varvec{a}\right.$$

$$\varvec{x}^{\varvec{n}} = \varvec{y}^{\varvec{n}} + \mathrm{DropPath}\left(\mathrm{LayerScale}\left(\mathrm{FFN}\left(\mathrm{LN}\left(\varvec{y}^{\varvec{n}}\right)\right)\right)\right) \quad (2)$$

Table 4. Transformer encoder configuration comparison.

Parameter	ViT-Base	HTR-VT	Ours
Encoder layers	12	4	8
Attention heads	12	6	8
Embedding dimension	768	768	768
FFN hidden dimension	3,072	3,072	3,072
Activation function	GELU	GELU	GELU
Positional embeddings	Learned	2D sinusoidal (fixed)	2D sinusoidal (fixed)
Class token	Yes	No	No
LayerScale	No	No	Yes (1×10^{-5})
Stochastic depth	No	No	Yes (0→0.1)
QKV bias	Yes	Yes	Yes
Patch embedding	Linear projection	CNN (ResNet-18)	CNN (ResNet-inspired)

As follows are the primary configuration decisions. The depth for the Sorani script model is doubled from the 4 layers in HTR-VT to 8, as it necessitates the resolution of multiple concurrent dependencies, including diacritic-to-base character association, four positional character forms, and flowing ligature continuity. These dependencies necessitate more successive attention passes than those in Latin-script HTR. We increase the number of attention heads from 6 to 8 to facilitate parallel specialization across these distinct relationship categories. For the sake of data efficiency, we implement fixed 2D sinusoidal positional embeddings [31] to circumvent the expense of learning positional parameters on a limited dataset. To stabilize the training of the deeper encoder,

we introduce LayerScale [32] with an initialization of 1×10^{-5}, which enables each block to progressively learn its contributing factor. We employ stochastic depth [33] with a linearly increasing drop rate of 0→0.1 to regulate the subsequent encoder blocks against overfitting. Therefore, no class token is employed, as the assignment necessitates that each spatial position contribute to the CTC output sequence.

4.3. CTC decoder

A linear projection is used to map each of the 128 output tokens from the 768-dimensional embedding space to K = 56 classes, which includes the CTC blank token and 55 Sorani characters and vocabulary in DASNUS, following the transformer encoder. The CTC loss [7] is employed for both inference and training, marginalizing over all valid alignment paths between the predicted and ground-truth sequences without necessitating explicit character segmentation. This property is particularly well-suited to connected cursive script. A clear evaluation of the model's intrinsic recognition capability is achieved by employing greedy decoding during inference in the absence of an external language model.

4.4. Training strategy

A summary of all hyperparameters is provided in Table 5. AdamW, which is augmented with SAM [34], is employed to optimize for flat loss landscape minima and to significantly enhance generalization on limited datasets [10]. Over 1,000 iterations, the learning rate increases linearly from 1×10^{-7} to 1×10^{-3}, and then decreases to zero over 100,000 total iterations, following a warm-up cosine annealing schedule. An EMA with a decay of 0.9999 is maintained for the purposes of evaluation and inference.

Table 5
Training hyperparameters.

Hyperparameter	Value
Optimiser	AdamW + SAM
Base LR	1×10^{-7} (warm-up start)
Max LR	1×10^{-3}
LR schedule	Warm-up cosine
Warm-up iterations	1,000
Total iterations	100,000
Weight decay	0.5
Batch size	8
AdamW betas	(0.9, 0.99)
EMA decay	0.9999
Hardware	1 × RTX 4090 (24 GB)

A span feature masking strategy [10] is implemented on the input token sequence prior to positional embeddings during training. This strategy replaces contiguous spans with a shared learnable mask token. We employ a mask ratio of $\tau = 0.3$ and a maximum span length of 4 to prevent the masking of entire words in the more densely packed Sorani script, which is smaller than the values of HTR-VT ($\tau = 0.4$, span = 8).

In Table 6, the stochastic data augmentation pipeline (p = 0.5 per augmentation) is described in detail. The input images are resized to 64×512 grayscale.

Table 6
Data augmentation pipeline with parameter ranges.

Augmentation Type	Parameter Range
Elastic Distortion	$\alpha \in [0.5, 1.0]$, $\sigma \in [1, 10]$, kernel ≤ 3
Perspective Warping	distortion ∈ [0.0, 0.4]
Morphological (Dilation/Erosion)	iterations = 1, kernel ≤ 3
Color & Contrast Jitter	brightness ± 0.4, contrast ± 0.4, hue ± 0.2, saturation ± 0.4
Gaussian Blur	kernel ∈ [3, 5], $\sigma \in [3, 5]$
Sharpening	strength ∈ [0, 1], alpha ∈ [0, 1]
Zoom (H/W scaling)	H ∈ [0.8, 1.0], W ∈ [0.99, 1.0]
Span Masking	mask ratio = 0.3, max span length = 4

4.5. Regularization strategy

Our model addresses the fundamental challenge of training a deep model on a limited cursive script dataset by employing a coordinated multi-level regularization strategy, as summarized in Table 7. Each of them operates at a separate stage of the pipeline, offering complementary rather than redundant protection against overfitting.

Table 7
Multi-level regularisation strategy.

Technique	Component	Purpose
Dropout (p = 0.3)	CNN feature extractor	Prevents co-adaptation of CNN features; reduces overfitting in the feature extraction stage
Span masking (τ = 0.3, s = 4)	Transformer input tokens	Forces contextual learning; prevents tokens from relying solely on local information
LayerScale (init = 1\times{10}^{-5})	Each transformer block	Stabilises training of the 8-layer encoder; enables gradual contribution learning
Stochastic depth (0→0.1)	Each transformer block	Regularises deeper blocks; prevents overfitting in later layers
Weight decay (0.5)	All parameters	L2 regularisation on model weights; prevents parameter magnitude growth
EMA (α = 0.9999)	Model-level	Temporal ensembling; smooths parameter trajectory for better generalisation
Data augmentation (p = 0.5)	Input images	Simulates real-world variation
SAM optimiser	Optimisation	Seeks flat minima in loss landscape; improves out-of-distribution (OOD) generalisation

5. Results and Experiments

To verify the performance of our HTR-VT adapted model, we conducted a comprehensive series of ablations on the DASNUS dataset. The ablations were aimed not only at measuring the training convergence and error rate of the model, but also its relative performance with respect to state-of-the-art HTR systems.

5.1. Progressive ablation study

In order to study the relative importance of each part in our model, we have done a progressive ablation analysis. Starting with a baseline ViT architecture with normal patch embeddings and a CTC decoder, we added architectural and learning refinements incrementally. At each step, CER and WER on the validation and test set were computed. Results have been given in Table 8.

Table 8
Main ablation (progressive build-up).

Variant	Validation		Test	
	CER (%)	WER (%)	CER (%)	WER (%)
Baseline (ViT patch embedding only)	5.30	27.10	5.10	26.20
+ CNN feature extractor	4.44	21.85	4.25	20.92
+ Span masking	4.21	20.82	4.07	19.94
+ Data augmentation	3.76	19.24	3.75	18.28
+ Deeper transformer (6 layers)	3.82	19.17	3.76	18.32
+ Deeper transformer (8 layers)	3.79	18.91	3.76	18.38
+ DropPath (0.1) + LayerScale (1e–5), 4 layers	3.69	18.73	3.67	18.17
+ DropPath (0.1) + LayerScale (1e–5), 8 layers	3.59	18.01	3.57	17.53

The base model achieved decent results but was not suitable for recognizing continuous strokes and ligatures common in the Sorani script, as it relied on patch embeddings. Replacing patch embeddings with our proposed feature extractor resulted in the highest improvement, as convolutional features are more suitable for preserving stroke characteristics and continuity. The model's performance was also improved using span-based masking and data augmentation. The former helped improve character reconstruction from incomplete or faded parts, while data augmentation, including distortions and contrast, improved model generalization for different styles of writing. The model's depth was increased, but it did not improve performance on test set and even resulted in slight overfitting due to the small size of the dataset. However, when using regularization techniques like DropPath and LayerScale, increasing depth was advantageous. An 8-layer model achieved the best results, demonstrating that the combination of all techniques, rather than any single technique, improved performance.

5.2. Isolated ablation study

While the cumulative improvements in section 5.1 demonstrate how successive modifications contribute to the final architecture, they do not reveal the independent effect of each factor. To address this, we conducted an isolated ablation study, starting from the best-performing progressive model in section 5.1 and altering one component at a time. This allows us to disentangle the role of training stabilizers, positional encodings, masking strategies, and architectural variations in the ResNet-inspired encoder. Results are presented in Table 9.

Table 9
Isolated ablation study (one factor at a time).

Variant	Validation		Test	
	CER (%)	WER (%)	CER (%)	WER (%)
No EMA	3.63	18.07	3.58	17.54
Optimizer = AdamW	3.83	18.90	3.80	18.40
Learnable positional enc.	3.73	18.75	3.62	17.84
Mask ratio = 0.15	3.72	18.59	3.76	18.53
Mask ratio = 0.45	3.74	18.83	3.64	18.21
Max span length = 1	3.87	19.34	3.80	18.67
Max span length = 8	3.73	18.22	3.57	18.00
No dilation	3.83	19.26	3.76	18.45
No pool after L2	3.72	18.63	3.69	17.98
Stride = (1,1)	3.58	18.18	3.47	17.37

The performance was slightly impaired by disabling EMA, indicating its importance in stabilizing the training and improving performance through generalization. The performance was impaired by a larger margin when the SAM optimizer was replaced by AdamW. The SAM optimizer performed better because it focuses on finding flatter minima of the loss function, making the model more robust to variations of handwritten text by different writers. The performance was impaired when deterministic sinusoidal positional encodings were replaced by learnable embeddings. The sinusoidal encodings performed better because they maintain deterministic positional consistency, which is important for the ligature-rich structure of Sorani handwritten text. Experiments conducted to find the best masking approach indicated that a moderate amount of masking was necessary. Less masking was not effective because it failed to induce contextual learning. Excessive masking, however, eliminated important information. The best performance was obtained when the span lengths were moderate because they are representative of the irregularities of handwritten text. The importance of the different components of the ResNet-inspired encoder was also identified. The dilation and max pooling components of the ResNet-inspired encoder were important. The performance was impaired when either of the two was disabled. The best performance was obtained when the initial stride was (1,1) instead of (2,1). The horizontal information was important in the early stages of the network, which was necessary to recognize subtle differences such as diacritics and spacing. The best-performing configuration was a combination of a well-tuned of our CNN feature extractor, an 8-layer Transformer network, regularization, sinusoidal positional encodings, balanced masking, the SAM optimizer, and dropout. The ablation study indicates that the different components of the Sorani HTR model, both major and minor, contributed to achieving stable and accurate performance.

This final setup was trained with 100,000 iterations and a scheduled learning rate. The learning rate was initially set at 0.0001 and stepped up to 0.001 during the initial 1,000 iterations so that early fast convergence was allowed (Fig. 6a). During training, both losses on the training and validation sets had smooth and continuous decreasing trends, with loss on the training set falling from 190.69 to 7.61 and loss on the validation set from 37.81 to 9.81 (Fig. 6b). This narrow and stable difference between these two curves also points to good generalization and little overfitting, supporting the stability in model configuration deduced in the ablation studies.

Figure 6c presents CER and WER, which also decreased with the passage of time. In iteration 1000, the CER was 19.76% and the WER was 67.74%, which decreased with the improvement of training. By iteration 50,000, the CER went below 5%, and the WER was in the mid-20%. Finally, the model had the best validation CER of 3.58% and the WER of 18.18%.

5.3. Comparison with state-of-the-art

This section compares the proposed HTR model with extant state-of-the-art systems across multiple datasets. Initially, we conducted a comparison between the proposed architecture and the HTR-VT model [CITE], which served as the foundation for our methodology. HTR-VT was re-implemented and trained on the DASNUS dataset under identical conditions to guarantee a fair comparison. The baseline achieved a validation CER of 4.16% and a test CER of 4.09%, with validation and test WERs of 20.46% and 19.80%, respectively. Conversely, our modified model demonstrated superior outcomes, attaining a validation CER of 3.58%, a test CER of 3.47%, a validation WER of 18.18%, and a test WER of 17.37%. These enhancements illustrate that the modifications introduced, including the ResNet-inspired encoder with stride adjustments, span masking, data augmentation, and optimized training strategies, considerably enhance the recognition of Sorani handwriting.

In order to evaluate the performance of the models for the task of generalization, the models were also tested on an additional set of 100 handwritten text lines from writers outside the DASNUS dataset. In this case, the performance of the re-implemented baseline model resulted in a CER of 7.57%, while the proposed model resulted in a CER of 6.59%. Similarly, the proposed model resulted in a WER of 31.76%, while the baseline model resulted in a WER of 35.89%. This shows that the proposed model has been revised to be more robust for the task. The results have been summarized in Table 10.

Table 10
Comparison of our adapted model with the re-implemented HTR-VT [10] on DASNUS and on an external set of 100 real-world Sorani text-lines. Our model consistently outperforms the baseline across both benchmarks.

Variant	Dataset	Validation		Test	
		CER (%)	WER (%)	CER (%)	WER (%)
HTR-VT [10] (baseline)	DASNUS	4.16	20.46	4.09	19.80
Ours (adapted model)	DASNUS	3.58	18.18	3.47	17.37
HTR-VT [10] (baseline)	External (100 lines)	–	–	7.57	35.89
Ours (adapted model)	External (100 lines)	–	–	6.59	31.76

Additionally, we conducted a comparison between our model and a variety of Arabic handwriting recognition methods on the KHATT dataset (Table 11). The proposed system obtained a validation CER of 7.97% and a test CER of 5.05% without the use of any external language model or lexicon. Our proposed model attained more robust character-level accuracy while functioning as a single visual model, despite the fact that an ensemble BLSTM–CTC system [35] reported a lower WER (13.52%) as a result of language modeling and output voting. In comparison to other systems, including MDLSTM-CTC [36], CNN–BLSTM–CTC [19, 37], and Transformer-based models [22], our proposed architecture obtained a significantly lower CER. This was primarily due to the CNN-Transformer design, which captures both local stroke features and long-range contextual dependencies. Improved computational efficiency, reduced dependence on sequential recurrence, and avoidance of handcrafted features are all advantages of this design.

Further evaluation was conducted on Katib's Pashto Text Imagebase (KPTI) [38], which showed strong cross-script generalization (Table 11). Our proposed model, based on the CNN, Transformer, and CTC, was found to have a validation CER of 3.04% and a test CER of 3.10%, which is a reduction of about 66% compared to the MDLSTM baseline [38]. Overall, the proposed model is shown to not only perform better than a strong baseline on the Sorani handwriting dataset but also perform state-of-the-art on a range of Arabic-script datasets, all without the need for any external linguistic resources.

Table 11
Performance comparison on the KHATT and KPTI datasets.

Dataset	Reference	Methods	CER (%)
KHATT	[35]	BLSTM + CTC	7.85%
	[36]	MDLSTM + CTC	19.98%
	[37]	CNN + BLSTM + CTC	19.85%
	[19]	ResNet–BiLSTM–CTC hybrid model + 3-gram LM	13.2%
	[22]	Transformer	18.45%
	Ours	ViT-based model	5.05%
KPTI	[38]	BLSTM and MDLSTM	9.22%
	Ours	ViT-based model	3.10%

5.4. Misrecognized character analysis

The character-level error analysis reveals numerous systematic defects in the model that are not apparent through aggregate CER and WER metrics alone. The model's propensity to omit characters, particularly spaces and faint letters, is one of the most significant issues. This results in merged words and structural distortions in the transcription. This behavior is plainly demonstrated in Fig. 7a, where the model fails to detect the inter-word space in "□□□□□□□□ □□□□□□□□□□," resulting in the entire phrase being merged into a single token. The DASNUS test set frequently contains such omissions due to the highly inconsistent spacing practices of handwritten Central Kurdish, which range from wide gaps to tightly connected letters that visually resemble ligatures.

Dot-based confusions between visually similar characters are another significant source of error. Letters such as □ ,□ ,□ ,□, and □ are primarily distinguished by the number or location of their dots, which are frequently overlooked in negligent or artistic handwriting styles. The most misrecognized characters presented in Fig. 8, as the confusion pairs □ ,□, and □ ,□ are among the most prevalent. This type of error is frequently the result of the dots being either indistinguishable or absent in the original handwriting, which renders the letters nearly identical during the handwriting process. A practical example of this type of error is illustrated in Fig. 7b, where the indistinguishable form of □ leads to the confusion of □□□□□□ with □□□□□□.

Another observation is that the model overpredicts; therefore, the model introduces characters that do not exist. This is attributed to the model's inability to differentiate between character strokes and noise. The inability of the model to recognize punctuation marks, especially dots, is an indication of the model's low sensitivity to small details. It is also evident from the observations that the model can easily miss a character or add an extra character even if the model is doing extremely well.

The aforementioned observations broadly classify the model's misrecognition of characters into three categories: overpredicted characters, dot-based confusions, and missing characters. Based on the aforementioned observations, it is evident that the model's misrecognition of characters is attributed to the complexity of Kurdish cursive handwriting and the model's inability. The model's accuracy can be improved by implementing the proposed solutions,

which include a better context sequence model to leverage the characteristics of Kurdish words, an enhanced attention model to increase sensitivity to small details, and the incorporation of dot recognition.

5.5. Discussion of results and findings

The performance obtained by our adapted HTR-VT model on the DASNUS dataset demonstrates a major improvement in the field of CKB handwriting recognition. The model not only generalizes very well on unseen text, with its CER of 3.47% and WER of 17.37% on the last test set, but also surpasses many of the current methods formulated for other cursive and semitic languages. Through these experiments, it can be inferred that certain architectural decisions along with the training process we used played a valuable role in enhancing the recognition performance, especially in reducing the number of character-level misclassifications.

One of the most important contributing elements to the low CER was the our HTR-VT architecture's design, which makes use of a vision-transformer backbone configured for the specific visual and spatial characteristics of Sorani handwriting. This is in contrast to standard CNN-based models that can be prone to the minute differences in curvature, shape, and ligature position typical of cursive scripts. The transformer architecture is well-suited to handle long-range relationships as well as fine-grained differences along sequences of visual tokens. This enabled the model to attempt to better discriminate among characters that are very similar in shape but varying in position, context, or small diacritics.

Another major reason for this performance is the quality and size of the DASNUS dataset. The DASNUS dataset, being a large dataset of Sorani handwritten text-line images, offered a wide range of ligatures of characters, handwriting styles, and inter-word space variability. The model was thus subjected to a wide range of writing styles. Furthermore, since the model was trained on the entire text-line images and not just the cropped characters, this approach helped to reduce confusion between morphologically ambiguous characters.

The strategy of training itself was also pivotal. Early on, aggressive learning rate (up to 0.001) helped the model quickly acquire global features of the handwriting. Later on, a gradual decay of the learning rate enabled the model to refine its weights subtlety, attending to fine details without deranging previously acquired patterns. This resulted in a smooth, stable decline of both training loss and validation loss, with very little overfitting, as indicated by the close correspondence of validation and test performance.

Our emphasis on CER in particular was intentional, since CER is a smaller-grained measure of recognition quality compared to WER, particularly in richly inflected languages such as CKB. In most real-world applications such as manuscript digitization or Optical Character Recognition (OCR)-based search, partial word recognition is useful even if most of the characters are correctly recognized. Thus, low CER is a good sign of usefulness in subsequent tasks such as indexing, archiving, or helping preserve languages.

Compared to models we studied in 5.4 Comparison with State-of-the-Art, we can see that our architecture fits better with the structure and linguistic requirements of Sorani handwriting. Other models with good performance on Latin or Arabic were found wanting as soon as they are applied to scripts with complicated joining behavior or varied stroke representations. Our HTR-VT's use of attention mechanisms and global receptive fields addresses this by allowing the model to acquire context-sensitive character embeddings. Lastly, we highlight that transformer-based models, in conjunction with sufficient data variance, customized preprocessing, and curriculum-based methods, can redefine the benchmark for HTR in lesser-represented languages such as Kurdish. These findings not only provide evidence of our HTR-VT model's viability but also are a strong justification for future work on deep-learn-based script-specific models as well as multilingual HTR.

While our writer-disjoint evaluation provides a fair measure of generalization to unseen handwriting styles, a limitation of the present study is the absence of an external OOD test set collected under different acquisition conditions (e.g., new writers, varied scanning devices, alternative prompts). Such an OOD benchmark would allow us to measure robustness under truly novel scenarios. We plan to address this in future work by expanding DASNUS with additional forms collected from different environments and contributors.

While our evaluation demonstrates strong performance under writer-disjoint splits of DASNUS, broader generalization analyses remain an important direction. In particular, zero-shot testing on unseen Kurdish handwriting (e.g., new page formats, layouts, or writing prompts), one-shot and few-shot writer adaptation using Parameter-Efficient Fine-Tuning (PEFT) techniques such as adapters or LoRA, and cross-script transfer (e.g., Arabic or Persian pretraining followed by Sorani adaptation) would provide stronger evidence of robustness and adaptability. These evaluations were not included in the present work due to resource and dataset limitations, but they represent natural next steps in future research. We anticipate that such studies would reveal how much transferable structure exists across cursive scripts and whether parameter-efficient fine-tuning can further improve writer adaptation in low-resource contexts.

6. Conclusion and Future Work

This paper proposed a novel deep learning method for Central Kurdish Language (CKB) handwriting recognition with a ViT-based model structure modified for the task. Addressing the long-standing challenge of low-resource script recognition, particularly for a linguistically dense and visually complex script like CKB, we developed the DASNUS dataset—a high-quality, diverse, and annotated collection of over 11,000 handwritten text-line images from 867 writers. The dataset is a groundbreaking achievement, as it is the largest public dataset that is specifically designed for Sorani handwriting recognition.

Our modified model, which features a convolutional backbone inspired by ResNet and a span-masked transformer encoder, exhibited a robust ability to learn the global contextual relationships and local stroke information of CKB handwriting. We were able to effectively address ligature, position-sensitive glyph, and handwriting variation issues by structuring the model with careful consideration of horizontal continuity and span-based regularization at training time.

Favorable results were obtained in these experiments, as the adapted model attained a CER of 3.47% and a WER of 17.37% on the held-out test set. These rates are almost identical to the validation results, which shows that the model generalizes the training data much more than it overfits. It is important to take these results into consideration. Though it doesn't make use of synthetic inputs nor of pretraining on enormous corpora, its accuracy still stands comparably to the state-of-the-art systems trained with other languages. That in itself testifies to the stability of the architecture—particularly in the context of low-resource settings—and belies the promise of ViTs in dealing with the idiosyncrasies of cursive script.

Looking ahead, A number of avenues suggest themselves:

- Multilinguality: Future releases may provide support to other Kurdish dialects—or even totally related scripts such as Pashto or Uyghur—by transferring knowledge across related languages.
- Synthetic Data Creation: There also exists future promise in creating artificial Central Kurdish handwriting samples, potentially with GANs or with style-transfer methods, to augment the model to generalize better across diversified handwriting systems.
- Writer Identification and Adaptation: In the light of DASNUS's population richness, the next step shall comprise writer-adaptive identification systems or a fusion of writer identification and handwriting recognition.
- Multimodal Fusion: Handwriting with either oral modality or auditory data could potentially allow for more complete transcription systems to preserve the Kurdish language.
- In this research, we have proposed a ViT-based model of CKB handwriting recognition and have presented the DASNUS dataset. The new system produces rivaling outcomes with no dependency on linguistic priors or auxiliary lexicons. In future research, we intend to incorporate a lightweight language model or lexicon-based decoder to minimize even more the WER and enhance linguistic normality. In addition, we intend to extend comparative analyses by incorporating newer transformer-based models in an effort to put our approach in more direct competition with newer state-of-the-art models. We also intend to investigate multilingual transfer learning and cross-pretraining scripts approaches to improve recognition robustness in other cursive scripts with limited resources.

Briefly, this work lays a strong foundation for Kurdish handwriting recognition with state-of-the-art deep learning approaches and offers seminal resources and approaches that can be extended by the research community. It is a significant step toward balanced coverage of low-resource languages for AI-driven document processing and pattern detection systems.

Declarations

Participant consent statement: All participants who contributed handwriting samples to the DASNUS dataset were informed about the purpose of the study and voluntarily agreed to participate. Written informed consent was obtained from the participants prior to data collection. The collected forms contained only demographic information and handwriting samples, and no personally identifying information is included in the published dataset or the study. The study procedures followed institutional research and ethical guidelines.

Competing interests:

The authors declare no competing interests.

Funding:

this research did not receive any funding.

Author Contribution

FARAEDWN M. SALIH led the study's design, data collection, methodology, deep learning implementation, experiments, software, and manuscript drafting. ABDULBASIT K. AL-TALABANI contributed to data collection, validated and analyzed results, supervised the research, and reviewed the manuscript. Both authors approved the final version.

Data Availability

The dataset used in this study is publicly available at: https://data.mendeley.com/datasets/xdj9f55rkm/1

References

1. Afkari-Fahandari, A., Shabaninia, E., Asadi-Zeydabadi, F., Nezamabadi-Pour, H.: A Comprehensive Survey of Transformers in Text Recognition: Techniques, Challenges, and Future Directions. ACM Comput. Surv. **58**(1–111), 42 (2025). https://doi.org/10.1145/3771273
2. Romein, C.A., Rabus, A., Leifert, G., Ströbel, P.B.: Assessing advanced handwritten text recognition engines for digitizing historical documents. Int. J. Digit. Humanit. **7**, 115–134 (2025). https://doi.org/10.1007/s42803-025-00100-0

3. Bamoki, M., Wady, S.H., Badawi, S.: Data Brief. **60**, 111533 (2025). https://doi.org/10.1016/j.dib.2025.111533 Holy Quran Kurdish Sorani translation dataset for language modelling
4. Abdalla, P.A., Shakor, M.Y., Ameen, A.K., Hassan, D.A.: Critical review of the model description in 'Kurdish handwritten character recognition using deep learning techniques'. Gene Expr. Patterns. **56**, 119399 (2025). https://doi.org/10.1016/j.gep.2025.119399
5. Abdalla, P.A., Qadir, A.M., Shakor, M.Y., Saeed, A.M., Jabar, A.T., Salam, A.A., Amin, H.H.H.: A vast dataset for Kurdish handwritten digits and isolated characters recognition. Data Brief. **47**, 109014 (2023). https://doi.org/10.1016/j.dib.2023.109014
6. Qader, A., Rashid, I.: T.A.: A Comprehensive Dataset of Complete Kurdish Handwritten Characters and Digits. 1, (2024). https://doi.org/10.17632/shny5k9f4b.1
7. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd international conference on Machine learning. pp. 369–376. Association for Computing Machinery, New York, NY, USA (2006)
8. Kizilirmak, F., Yanikoglu, B.: CNN-BiLSTM model for English Handwriting Recognition: Comprehensive Evaluation on the IAM Dataset, (2023). http://arxiv.org/abs/2307.00664
9. Chan, A., Mijar, A., Saeed, M., Wong, C.-W., Khater, A.: HATFormer: Historic Handwritten Arabic Text Recognition with Transformers, (2025). http://arxiv.org/abs/2410.02179
10. Li, Y., Chen, D., Tang, T., Shen, X.: HTR-VT: Handwritten text recognition with vision transformer. Pattern Recogn. **158**, 110967 (2025). https://doi.org/10.1016/j.patcog.2024.110967
11. Pechwitz, M., Maddouri, S.S., Märgner, V., Ellouze, N., Amiri, H.: IFN/ENIT - DATABASE OF HANDWRITTEN ARABIC WORDS. In Proc. of CIFED. 127–136 (2002)
12. Mahmoud, S.A., Ahmad, I., Al-Khatib, W.G., Alshayeb, M., Tanvir Parvez, M., Märgner, V., Fink, G.A.: KHATT: An open Arabic offline handwritten text database. Pattern Recogn. **47**, 1096–1112 (2014). https://doi.org/10.1016/j.patcog.2013.08.009
13. Safarzadeh, V.M., Jafarzadeh, P.: Offline Persian Handwriting Recognition with CNN and RNN-CTC. In: 2020 25th International Computer Conference, Computer Society of Iran (CSICC). pp. 1–10 (2020)
14. Hassan, S., Irfan, A., Mirza, A., Siddiqi, I.: Cursive Handwritten Text Recognition using Bi-Directional LSTMs: A Case Study on Urdu Handwriting. In: 2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML). pp. 67–72 (2019)
15. Mohammed, T.L., Ahmed, A.A., Al-Sanjary, O.I.: KRDOH: Kurdish Offline Handwritten Text Database. In: 2019 IEEE 7th Conference on Systems, Process and Control (ICSPC). pp. 86–89 (2019)
16. Ahmed, R.M., Rashid, T.A., Fatah, P., Alsadoon, A., Mirjalili, S.: An extensive dataset of handwritten central Kurdish isolated characters. Data Brief. **39**, 107479 (2021). https://doi.org/10.1016/j.dib.2021.107479
17. Alsaqi, I.M., Fattah, P.: Kurdish Handwritten Word Recognition withCRNN Using Multi-Backbone Evaluation: Kurdish Handwritten Word Recognition. Acad. J. Int. Univ. Erbil. **2**, 439–449 (2025). https://doi.org/10.63841/iue24539
18. Maalej, R., Kherallah, M.: Convolutional Neural Network and BLSTM for Offline Arabic Handwriting Recognition. In: 2018 International Arab Conference on Information Technology (ACIT). pp. 1–6 (2018)
19. Mutawa, A.M., Allaho, M.Y., Al-Hajeri, M.: Machine Learning Approach for Arabic Handwritten Recognition. Appl. Sci. **14**, 9020 (2024). https://doi.org/10.3390/app14199020
20. Ahmed, R.M., Rashid, T.A., Fattah, P., Alsadoon, A., Bacanin, N., Mirjalili, S., Vimal, S., Chhabra, A.: Kurdish Handwritten character recognition using deep learning techniques. Gene Expr. Patterns. **46**, 119278 (2022). https://doi.org/10.1016/j.gep.2022.119278
21. Mostafa, A., Mohamed, O., Ashraf, A., Elbehery, A., Jamal, S., Khoriba, G., Ghoneim, A.S.: OCFormer: A Transformer-Based Model For Arabic Handwritten Text Recognition. In: 2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC). pp. 182–186 (2021)
22. Momeni, S., BabaAli, B.: A transformer-based approach for Arabic offline handwritten text recognition. SIViP. **18**, 3053–3062 (2024). https://doi.org/10.1007/s11760-023-02970-9
23. Kass, D., Vats, E.: AttentionHTR: Handwritten Text Recognition Based on Attention Encoder-Decoder Networks. In: Uchida, S., Barney, E., Eglin, V. (eds.) Document Analysis Systems, pp. 507–522. Springer International Publishing, Cham (2022)
24. Al-Ma'adeed, S., Elliman, D., Higgins, C.A.: A data base for Arabic handwritten text recognition research. In: Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition. pp. 485–489 (2002)
25. Mezghani, A., Kanoun, S., Khemakhem, M., Abed, H.E.: A Database for Arabic Handwritten Text Image Recognition and Writer Identification. In: 2012 International Conference on Frontiers in Handwriting Recognition. pp. 399–402 (2012)
26. Goh, T.Y., Basah, S.N., Yazid, H., Aziz Safar, M.J., Ahmad Saad, F.S.: Performance analysis of image thresholding: Otsu technique. Measurement. **114**, 298–307 (2018). https://doi.org/10.1016/j.measurement.2017.09.052
27. Li, S., Shen, Q., Sun, J.: Skew detection using wavelet decomposition and projection profile analysis. Pattern Recognit. Lett. **28**, 555–562 (2007). https://doi.org/10.1016/j.patrec.2006.10.002
28. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. Presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
29. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. Presented at the International Conference on Learning Representations October 2 (2020)

30. Xiong, R., Yang, Y., He, D., Zheng, K., Zheng, S., Xing, C., Zhang, H., Lan, Y., Wang, L., Liu, T.: On Layer Normalization in the Transformer Architecture. In: Proceedings of the 37th International Conference on Machine Learning. pp. 10524–10533. PMLR (2020)
31. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked Autoencoders Are Scalable Vision Learners. Presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022)
32. Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., Jégou, H.: Going Deeper With Image Transformers. Presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision (2021)
33. Huang, G., Sun, Y., Liu, Z., Sedra, D., Weinberger, K.Q.: Deep Networks with Stochastic Depth. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) Computer Vision – ECCV 2016, pp. 646–661. Springer International Publishing, Cham (2016)
34. Foret, P., Kleiner, A., Mobahi, H., Neyshabur, B.: Sharpness-Aware Minimization for Efficiently Improving Generalization, (2021). http://arxiv.org/abs/2010.01412
35. Jemni, S.K., Kessentini, Y., Kanoun, S., Ogier, J.-M.: Offline Arabic Handwriting Recognition Using BLSTMs Combination. In: 2018 13th IAPR International Workshop on Document Analysis Systems (DAS). pp. 31–36 (2018)
36. Ahmad, R., Naz, S., Afzal, M., Rashid, S., Liwicki, M., Dengel, A.: A Deep Learning based Arabic Script Recognition System: Benchmark on KHAT. IAJIT. **17**, 299–305 (2020). https://doi.org/10.34028/iajit/17/3/3
37. Lamtougui, H., Moubtahij, E., Fouadi, H., Satori, H.: An Efficient Hybrid Model for Arabic Text Recognition. Computers Mater. Continua. **74**, 2871–2888 (2023). https://doi.org/10.32604/cmc.2023.032550
38. Ahmad, R., Afzal, M.Z., Rashid, S.F., Liwicki, M., Breuel, T., Dengel, A.: KPTI: Katib's Pashto Text Imagebase and Deep Learning Benchmark. In: 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR). pp. 453–458 (2016)

Footnotes

1. https://data.mendeley.com/datasets/xdj9f55rkm/1

Figures

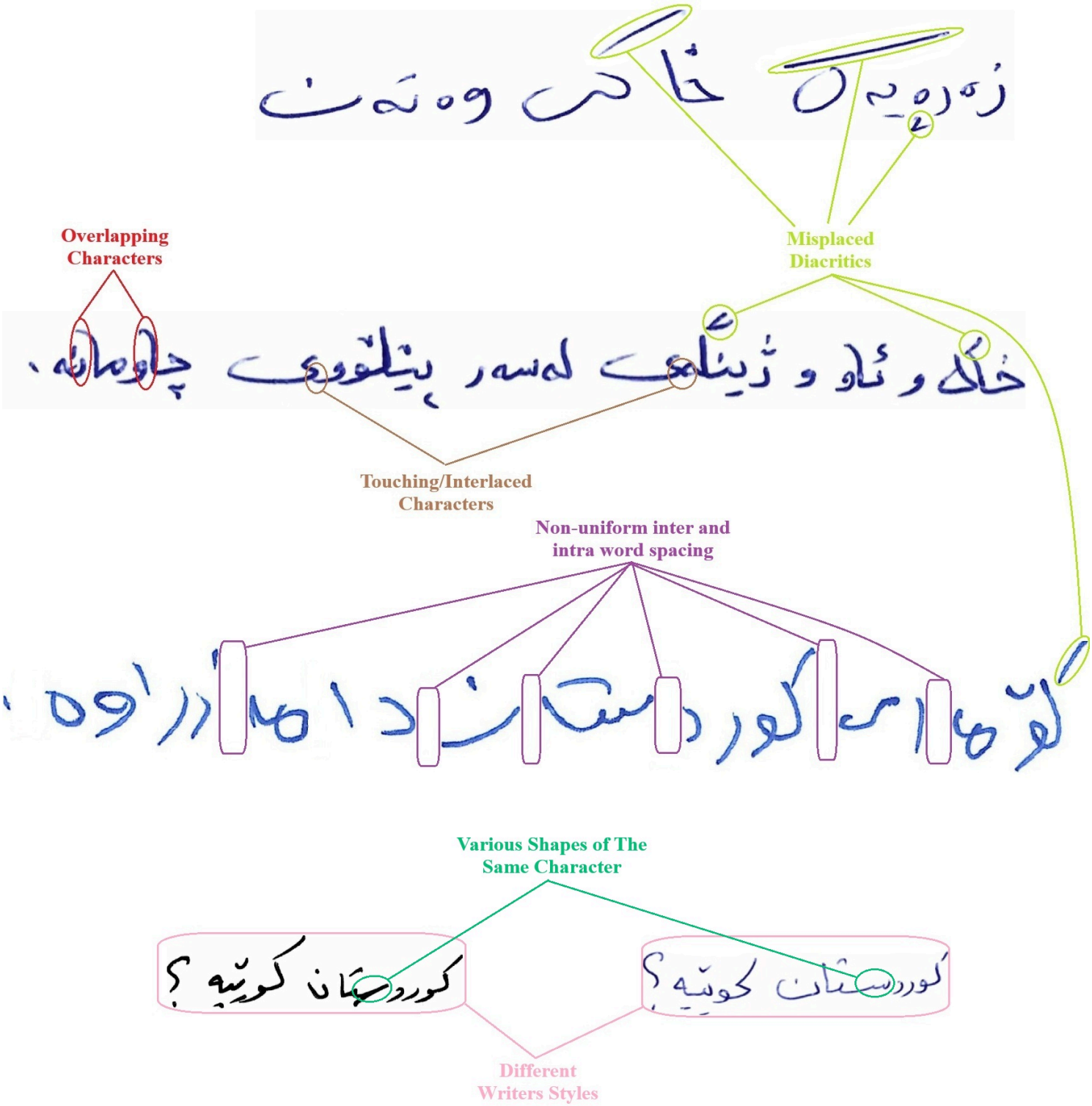


Figure 1

Common challenges of Sorani cursive handwriting.

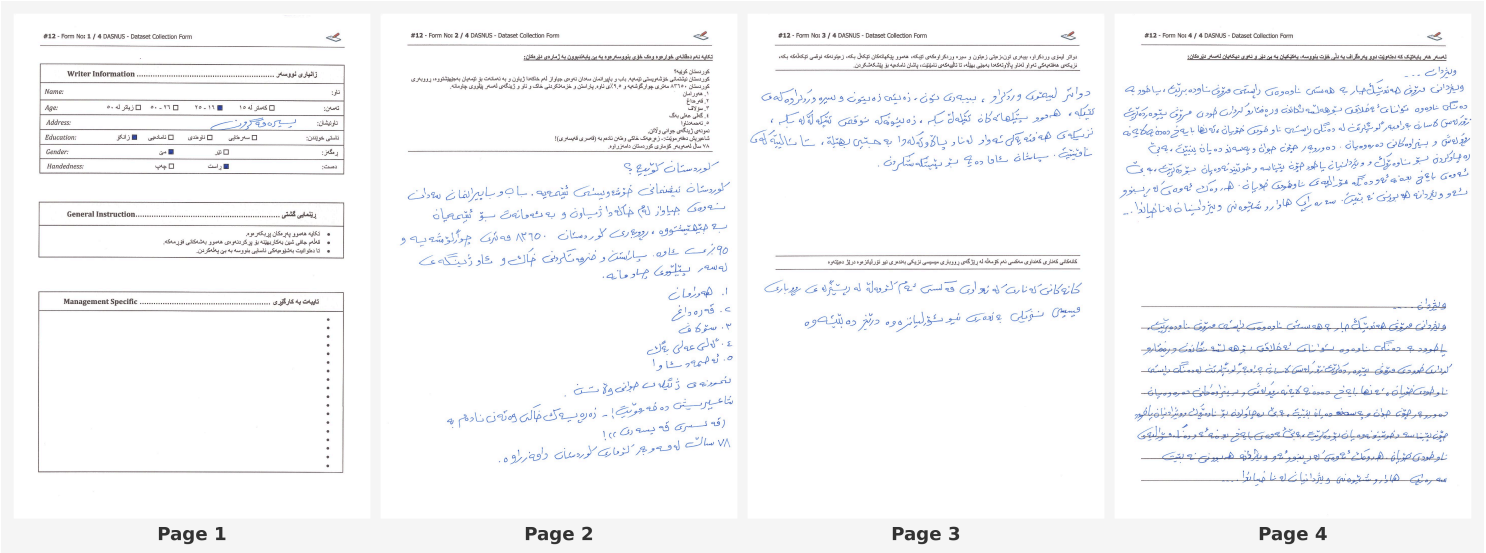


Figure 2

A representative sample of the DASNUS handwriting data collection form. Each form consists of four pages: the first records writer demographic information, while the remaining three pages contain Kurdish text prompts that participants were instructed to transcribe by hand using a pen in a their own writing style.

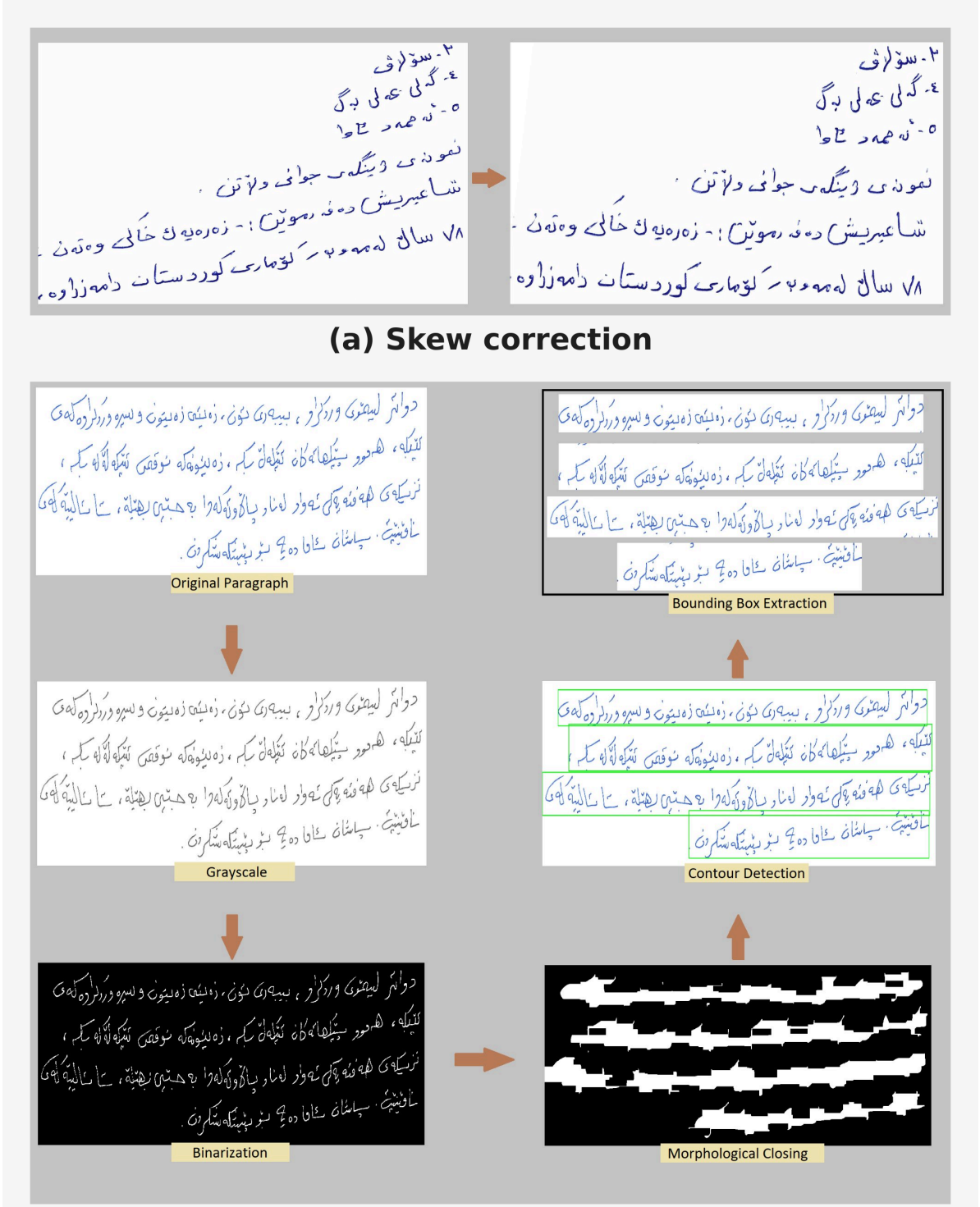


Figure 3

(a) Skew correction applied to incorrectly skewed paragraphs. (b) Text-line segmentation steps from a handwritten paragraph sample.

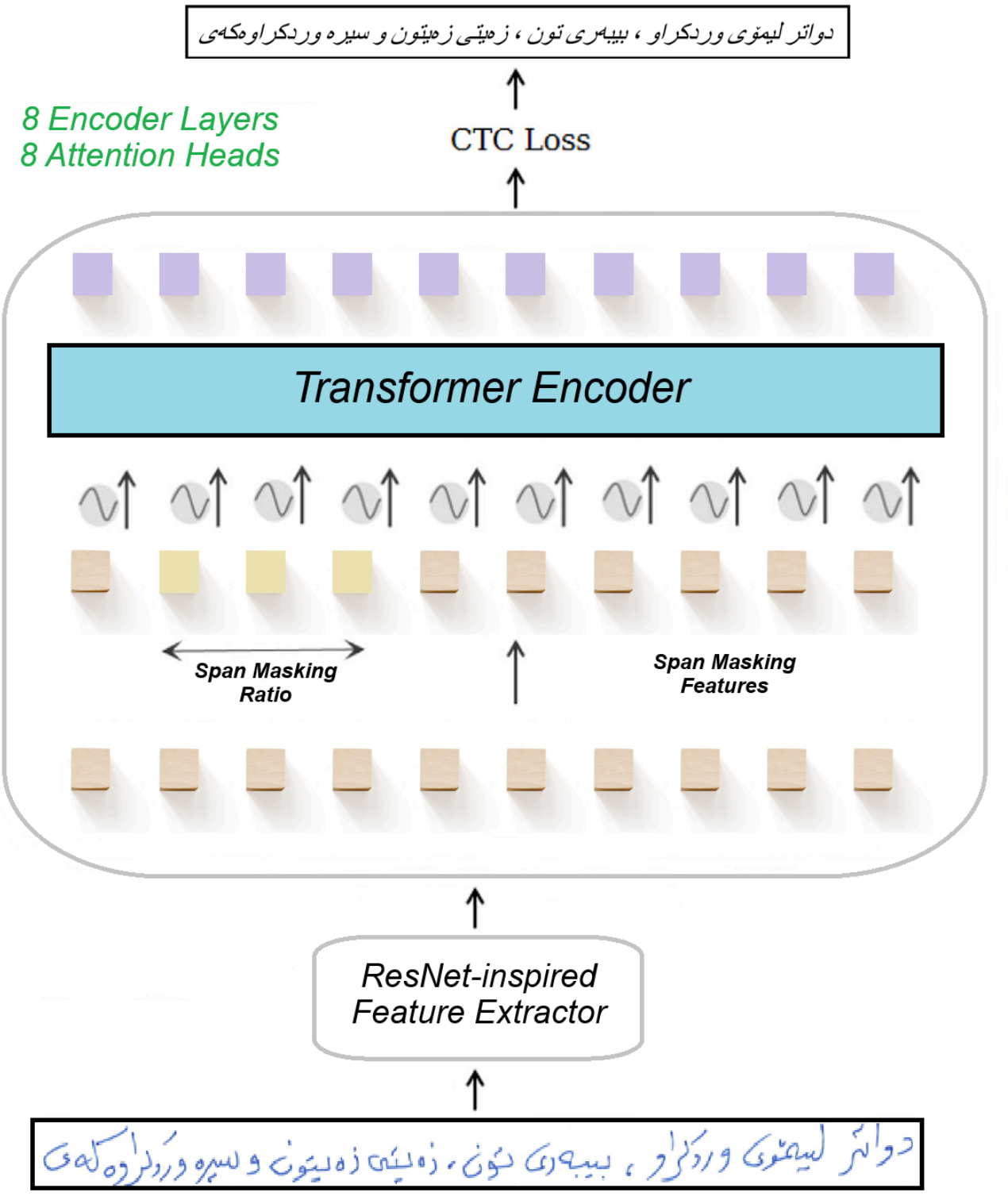


Figure 4

Our modified ViT-based model architecture.

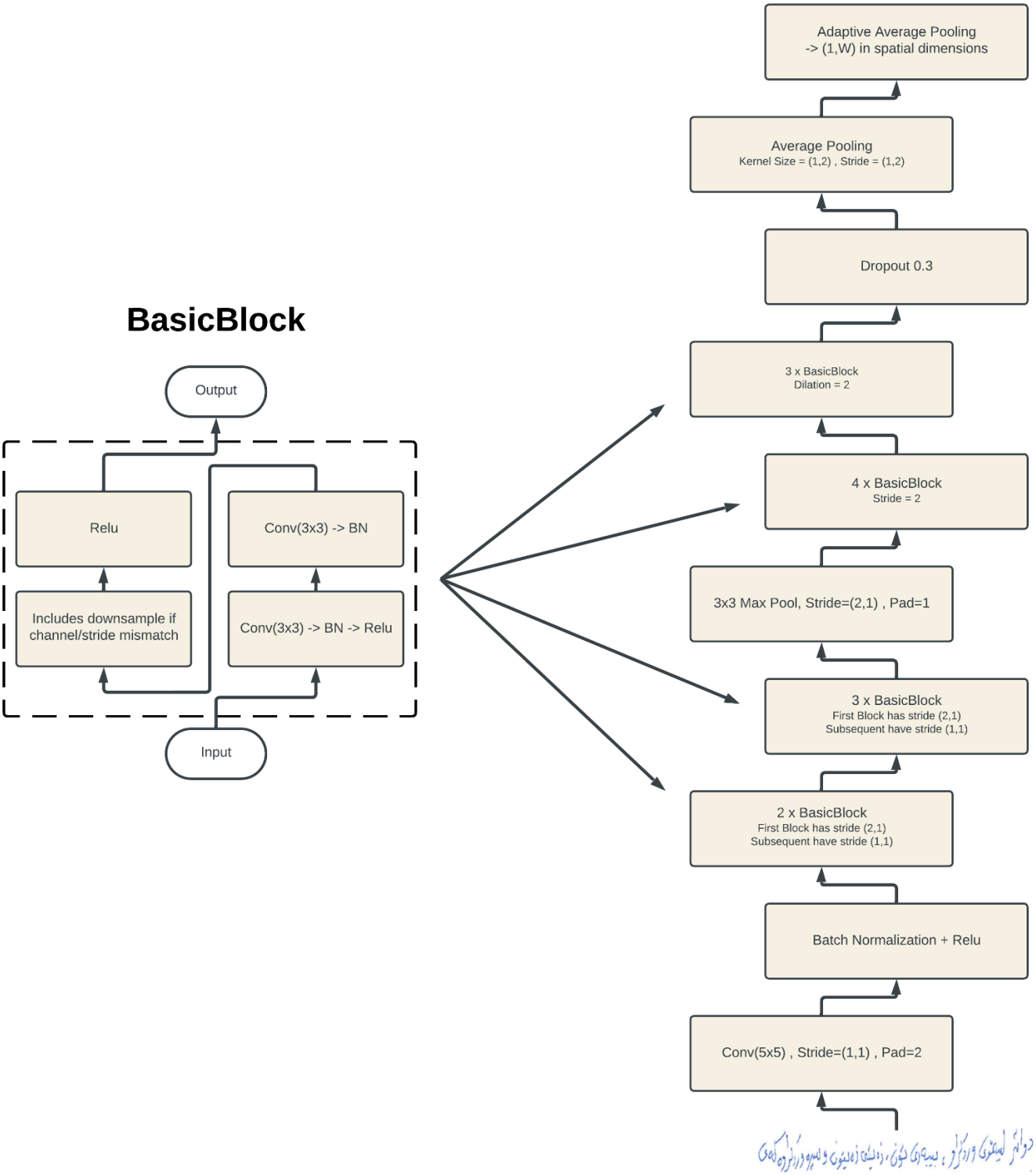
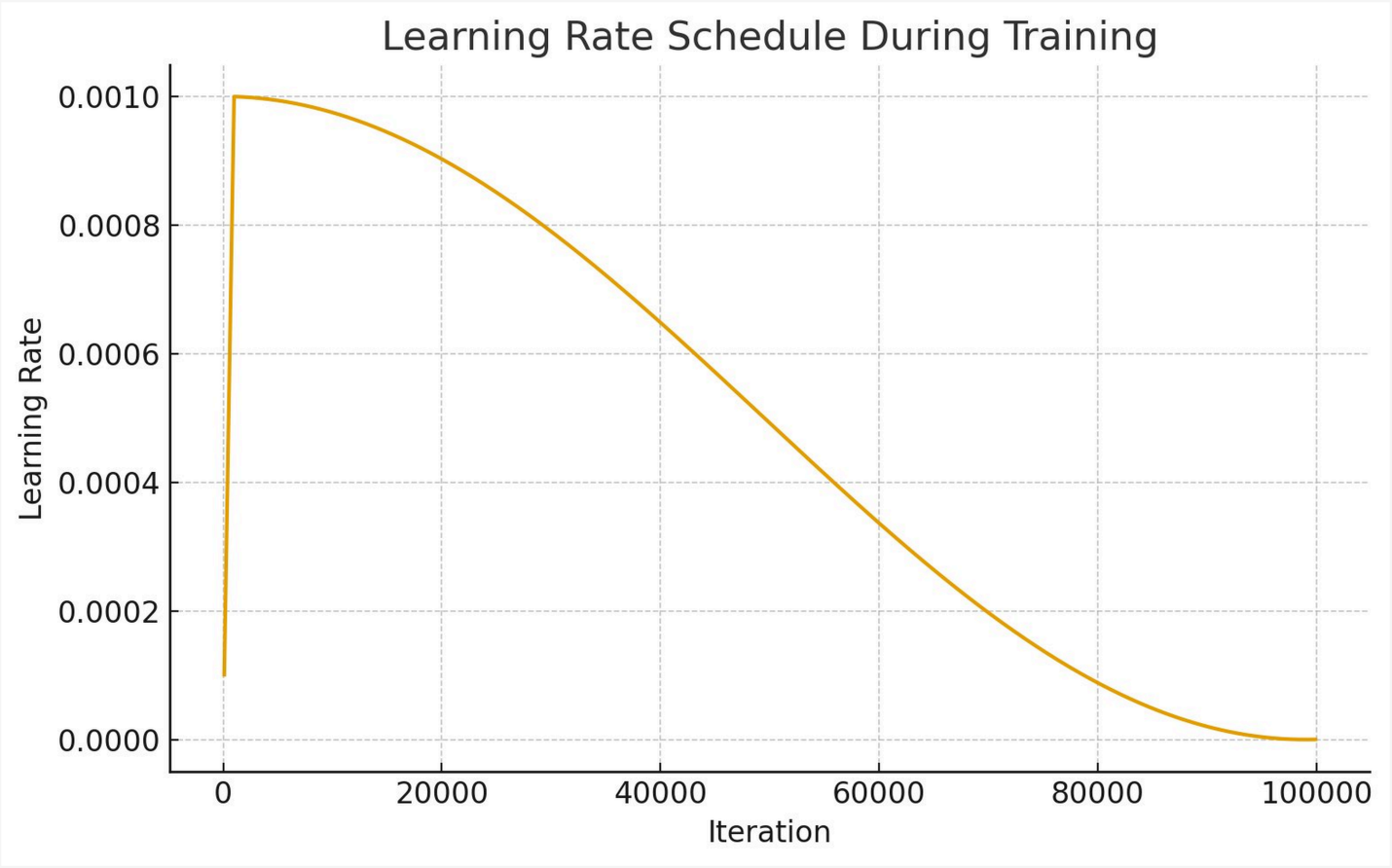
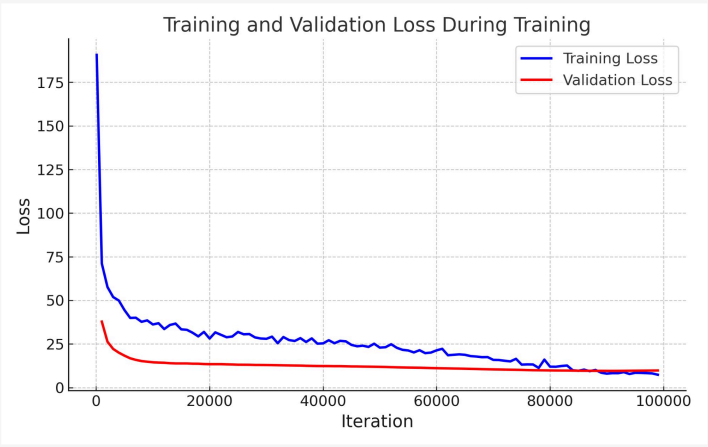


Figure 5

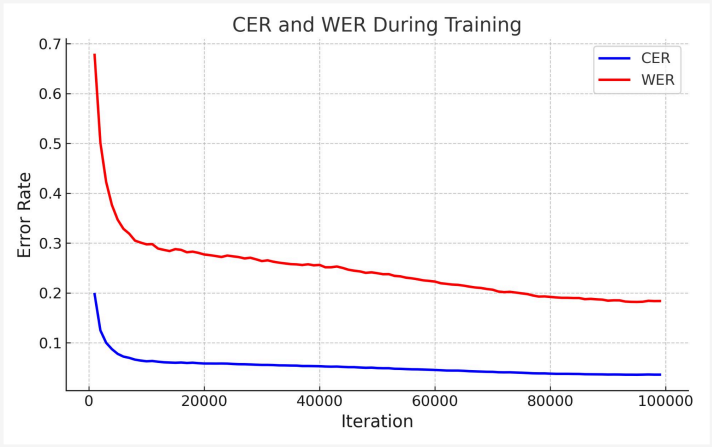
Our ResNet-inspired CNN feature extractor architecture.



(a) Learning rate schedule



(b) Training and validation loss



(c) CER and WER during training

Figure 6

(a) Learning rate schedule. (b) Training and validation loss functions. (c) Performance of the model in CER and WER during training and validation progress.



Figure 7

Some recognition examples from the DASNUS test set, showing the handwritten image, ground-truth transcription (green), and model prediction (red) for three samples. (a) (CER = 5.56%, WER = 40.00%): inter-word space omission merges "□□□□□□□□□□ □□□□□□□" into a single token. (b) (CER = 12.07%, WER = 70.00%): multiple diacritic substitutions and a spurious character insertion produce the highest error rate among the examples.

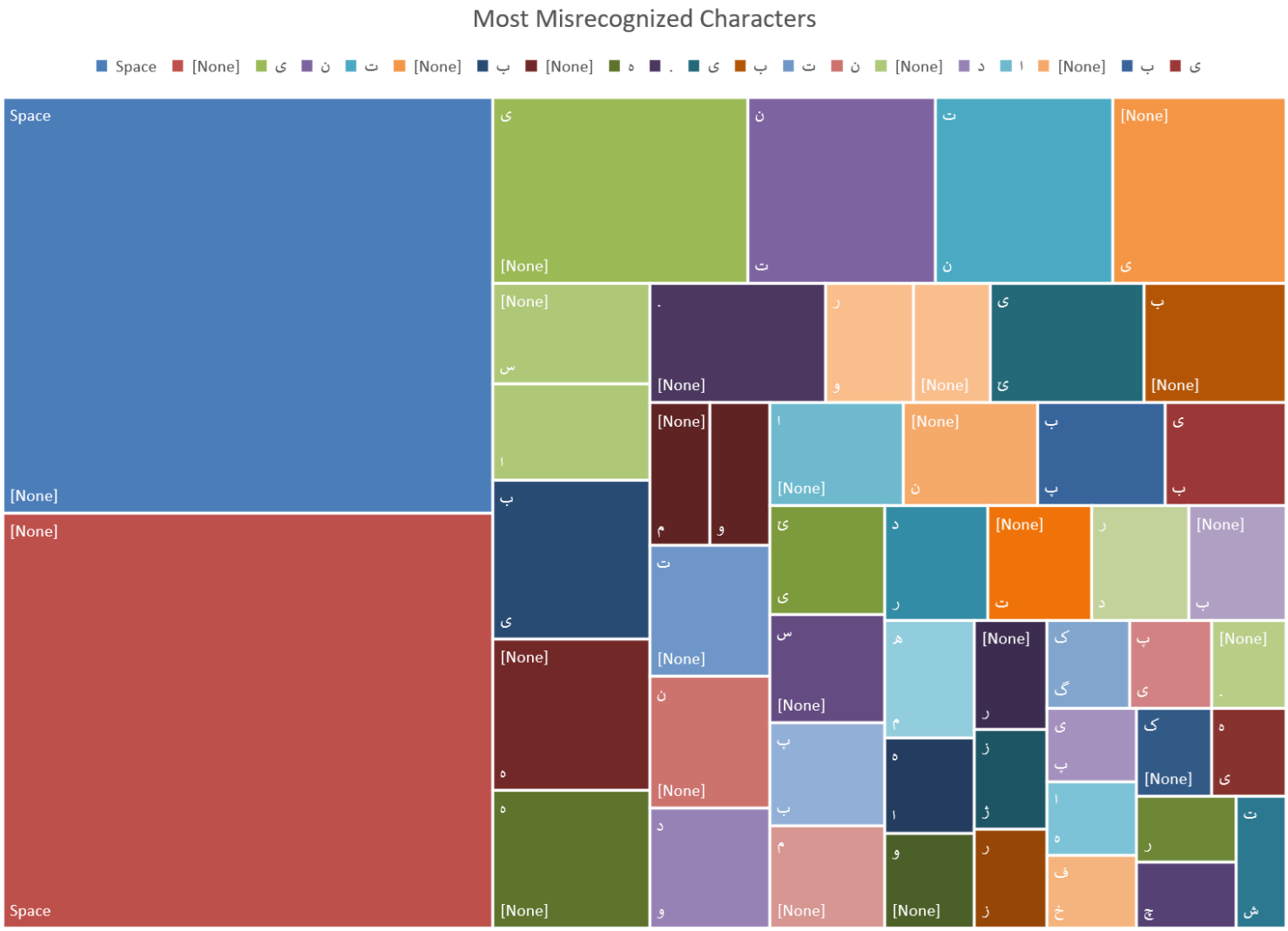


Figure 8

Most misrecognized characters.