KEC Conference 2026

ISSN: 3021-940X (Print) DOI: 10.3126/injet.v3i2.95148

Hybrid CRNN with Seq2Seq Attention Mechanism for Handwriting Recognition

Bhagwan Prasai[1], Avishek Gautam[2], Shruti Sapkota[3], Biniv Maharjan[4], Nirajan Acharya[5*]

[1]*Department of Computer and Electronics Engineering, Kantipur Engineering College, Dhapakhel, Lalitpur Nepal,* bhagwan.prasai444@gmail.com

[2]*Department of Computer and Electronics Engineering, Kantipur Engineering College, Dhapakhel, Lalitpur Nepal* avi.gtm14@gmail.com

[3]*Department of Computer and Electronics Engineering, Kantipur Engineering College, Dhapakhel, Lalitpur Nepal* shrutisapkota97@gmail.com

[4]*Department of Computer and Electronics Engineering, Kantipur Engineering College, Dhapakhel, Lalitpur Nepal* binivmaharjan21@gmail.com

[5]*Department of Computer and Electronics Engineering, Kantipur Engineering College, Dhapakhel, Lalitpur Nepal* nirajanacharya@kec.edu.np

Abstract

Handwritten Text Recognition (HTR) faces significant challenges including limited annotated data, high handwriting variability, and complex character formations. This paper proposes a hybrid CRNN with Seq2Seq Bahdanau attention for robust HTR. The encoder employs a ten-layer CNN with residual connections for spatial feature extraction and Bidirectional LSTM for temporal modeling. The decoder uses Bahdanau attention to dynamically generate context vectors by focusing on relevant image regions at each decoding step, combining character embeddings and context vectors to produce character probabilities via SoftMax. To prevent overfitting with limited data, comprehensive regularization is applied: dropout (0.3), weight decay (1e-4), label smoothing (0.1), stochastic depth (0.1), exponential moving average (EMA), stochastic weight averaging (SWA), adaptive teacher forcing decay, cosine annealing, and layer-wise learning rate decay, supplemented by light augmentation. Evaluation on the IAM handwriting dataset demonstrates 13.59% WER and 4.28% CER (86.41%-word level accuracy, 95.72%-character level accuracy), outperforming recent comparable CRNN-based methods. Attention visualizations confirm meaningful spatial-sequential alignment, with diagonal attention patterns indicating systematic left-to-right character progression. These results validate that attention-based sequence modeling combined with systematic regularization achieves robust HTR performance in data-limited scenarios without relying on synthetic data or external lexicons.

Keywords: Handwritten text recognition, CNN-BiLSTM-Attention Hybrid Model, Text recognition

1. Introduction

Handwritten Text Recognition has emerged as an essential technology with many applications in various fields. Historical documents, manuscripts and handwritten records all contain important information about culture, history, and scientific development. However, the text documents are mostly in physical form and hence vulnerable to environmental degradation. Digitalizing these handwritten texts has become an essential need to preserve important information and make it accessible for future research. However, digitalization and recognition of these handwritten texts remain a challenging problem.

Handwritten Text Recognition (HTR) aims to convert handwritten text images into digital text. However, handwritten text possesses variability in handwriting styles and the original documents may be in a poor state. These issues make Handwritten Text Recognition more challenging. These challenges suggest the need for more enhanced HTR systems.

Significant advances have been reported in the recent application of deep learning techniques to improve

the performance of HTR systems[9]. Current state-of-the-art approaches have been dominated by deep learning techniques which typically require significant amounts of training data. Neural network

* *Corresponding author*

architecture that combines Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have proven to be more effective in extracting spatial and sequential features. Convolutional Neural Networks extract visual features while Recurrent Neural Networks models the sequential dependencies between characters in a word [1]. Prior research has explored several deep learning techniques to improve performance of HTR systems. These methods can be categorized into two main groups: segmentation-based methods[2] and seq2seq-based approaches[3][5]. In an effort towards designing a data driven HTR pipeline, a Handwritten Text Recognition benchmark model [5] is studied, which is trained on more than 200,000-word images.

In this study, drawing inspiration from the effectiveness of encoder-decoder architectures, we propose an Attention-based Convolutional Recurrent Neural Network (CRNN) for handwritten text recognition. Our architecture consists of an encoder comprising CNN for visual feature extraction and bi-directional long short-term memory(BiLSTM) network to model sequential dependencies of characters. In the final transcription, a prediction block that uses Bahdanau Attention along with an LSTM is used, enabling the system to focus on certain spatial features before generating sequence of characters via SoftMax output.

The key contributions of our work can be summarized as follows: (1) We propose an end-to-end CRNN architecture integrating Bahdanau attention within a full seq2seq framework with a dedicated LSTM decoder, enabling explicit spatial attention at each decoding step rather than relying on CTC implicit alignment. (2) We apply Stochastic Depth (DropPath) within the CNN residual blocks a regularization strategy not previously applied in attention-based HTR encoders. (3) We introduce a combined training regime of Exponential Moving Average (EMA), Stochastic Weight Averaging (SWA), and adaptive teacher forcing decay that addresses overfitting from complementary angles without requiring synthetic data. (4) We introduce dynamic attention masking derived from actual image widths to prevent the attention mechanism from attending to padded regions in variable-width batches.

2. Literature Review

2.1. Related Works

The deep learning framework for handwritten text recognition has evolved significantly, with research branching into high-accuracy, revolutionary classification systems and in recent time a critical shift on the transformer model for better computational efficiency in real-world deployment. Teslya et al. (2022) provided a comprehensive survey of these deep learning approaches, highlighting key challenges such as data scarcity, style variability, and computational constraints [9].

Refining the approach Kass proposed an attention-based encoder-decoder network for the purpose of handwritten text recognition. [1] The proposed architecture has ResNet for feature extraction, two-layer BiLSTM for sequence modelling which makes up the encoder block, content-based attention and a LSTM block with SoftMax for the decoder part. The model was tested on both Imgur5k and IAM dataset and produced a CER of 6.50% and WER of 15.40%. Also, in the paper the author used transfer learning from scene text recognition to handwritten text recognition to address the problem of insufficient training data. Similarly, Kumari proposed a lexicon-guided attention-based HTR system, demonstrating that constrained decoding can further reduce recognition errors [2].

In the domain of scene text recognition Alshawi presented an attention-based convolutional recurrent neural network(CRNN). [3] Their study involved employing two-head CNN architecture augmented with SE(Squeeze-and-excitation) gates to enhance channel wise feature representation, followed by a Bidirectional Gated Recurrent Unit (Bi-GRU) for sequential dependency modeling and a CTC(connectionist Temporal Classifier) layer for a variable length sequence decoding. They compiled a dataset of 20,000 images of scene text and the proposed different model consisting of Bi-GRU, simple GRU, Bi-LSTM, simple LSTM and with Mobile Net and Inception, among them the model with Bi-GRU got the highest accuracy of them all with 94.26%.

The paper by Sunori focuses on using deep learning hybrid CNN-RNN architecture to advance offline handwritten text recognition [4]. They crafted a novel hybrid CNN-RNN model that was trained on an IAM dataset, composed of 87,292 training images and 4,316 testing images. The dataset was processed and segmented from where the text is extracted using CNN and the RNN then employs LSTM or GRU which help the model to predict what the next character should be in the handwritten text. The average accuracy of the proposed model was 80.33% with loss of 0.0116. Retsinas further explored enhancing CRNN architectures by incorporating Transformer blocks, demonstrating improved contextual modeling over standard recurrent layers [5].

Research by Jain proposed a method using CNN-BiLSTM along with CTC for medical prescription recognition [6]. The paper used CNN for feature extraction, BiLSTM for making predictions of each frame of the context vector with a linear layer and the final decoding stage to translate each character on the recognized LSTM layer into an alphabetic character using CTC loss function. They also built a corpus manually containing the widely used terms in the medical domain to make the predictions more aligned to the medical field. The proposed model performed well with CER of 0.0211 in training and CER of 0.0889 for testing on IAM line dataset. Similar work by Yadav applied region-based CRNN with CTC for medical handwritten text, further confirming the effectiveness of CRNN pipelines in domain-specific scenarios [7].

Truc proposed an HTR-ConvText, a model designed to capture the fine-grained stroke level feature while preserving global contextual dependencies [8]. The proposed model employs a novel Textual Context Module during the training phase. This module injects bidirectional textual context into a hybrid CNN-Transformer Visual encoder via an auxiliary loss, effectively enhancing the model's semantic capabilities while retaining the fast inference speeds characteristic of CTC-based systems.

Ali used CNN with SVM classifier with dropout for Arabic handwritten script recognition that automates both feature extraction and classification [9]. The use of Maximum Margin and Minimum classification error(M3CE) training rule which improves upon the standard cross-entropy often used in seq2seq. Evaluated across multiple Arabic datasets(AHDC,HACDB,IFN/ENIT, and AHDB) the model achieved exceptional accuracies ranging from 98.58% to 99.85%.

2.2. Research Gap

Handwritten text recognition systems face two practical challenges directly relevant to this work: (1) overfitting when training on limited annotated data, and (2) generalization across diverse handwriting styles within a single writer population. While stronger transformer-based systems exist, they typically require large-scale pre-training on hundreds of millions of synthetic and real text images, making them impractical in resource-constrained settings. Existing CRNN-based approaches with CTC decoding, though lightweight, rely on implicit sequence alignment that is difficult to interpret or verify. This work demonstrates that a systematically regularized attention-based seq2seq CRNN combining DropPath, EMA, SWA, and adaptive teacher forcing can achieve competitive recognition performance on the English IAM dataset across 657 writers without synthetic data augmentation. No claims are made regarding cursive scripts, diacritical marks, or cross-lingual generalization, which remain open problems beyond the scope of this study.

3. Methodology

This section describes the complete experimental framework used in this study which can be seen in figure 1.

It begins with IAM handwritten dataset and its preprocessing through resizing and augmentation, the CRNN architecture enhanced with Seq2Seq and Bahdanau attention, and the training parameters comprising the loss function, optimizer, and regularization techniques used in experiments.

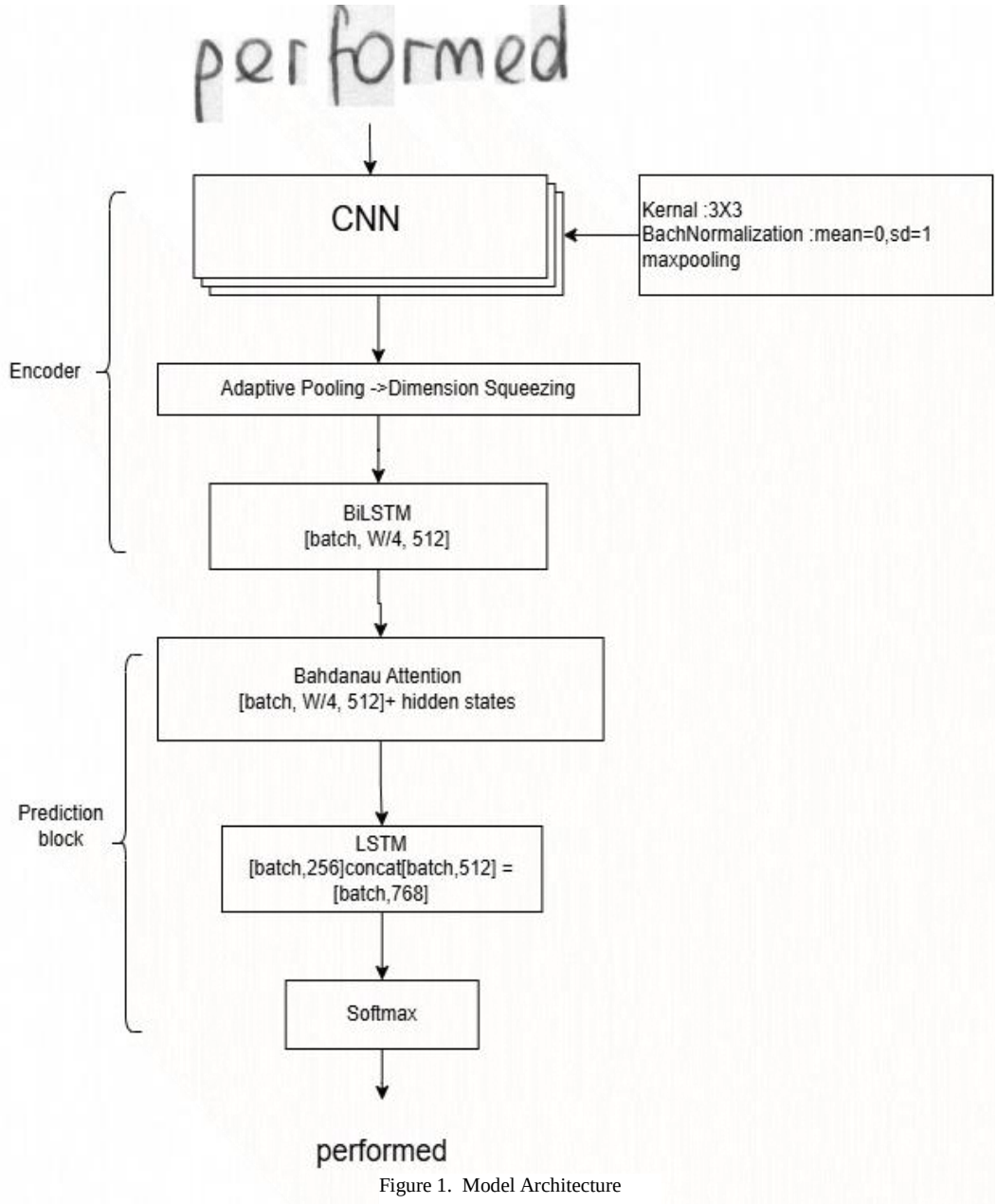

Figure 1. Model Architecture

2.3. Dataset Description

The IAM Handwriting dataset was first introduced at the ICDAR 1999 international conference, focusing on document analysis and recognition. Since then, this dataset has gone through several changes. The study uses version 3.0 of the IAM Handwriting dataset. The dataset contains handwritten samples from 657 writers, comprising 1539 pages of scanned text, 5685 isolated and labeled sentences, 13,353 isolated and labeled text lines and 115320 isolated and labeled words. For this study, we specifically utilized the isolated word subset which consist of 115320 labeled word images. The images show a variety of sizes, ranging from 1X1 to 116X1934 pixels, as well as differences in focus and illumination. To enhance the dataset different data processing techniques were applied like grayscale conversion, aspect ratio preservation, dynamic padding(in batch collection), gaussian blur. The dataset is further cleaned by

and some light augmentation techniques are applied during training for better generalization. After cleaning, 13,776 samples were removed, resulting in 101,544 usable word images. Following the cleaning process, the dataset is split into 71081 training words, 15,232 validation word and 15,231 as test words which can be seen in figure 2.

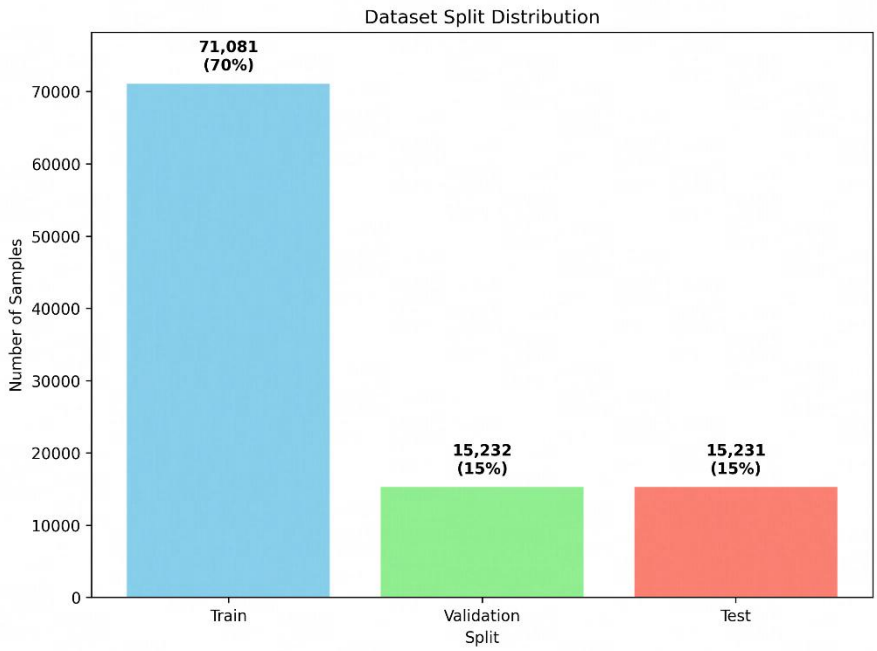

Figure 2. Dataset Split Distribution

2.4. Architecture Design

To implement the architecture designed to convert handwritten text into digital text. The framework is known as Convolutional Recurrent Neural Network (CRNN) architecture integrated within the Sequence-to-Sequence(seq2seq) model. It combines three stages: visual e*x*traction, sequential modeling, and attentive decoding. Convolutional Recurrent Neural Network(CRNN)-attention pipeline the given fig1 illustrates the end-to-end process of converting a handwritten image into digital text. The architecture is divided into functional stages:

2.4.1. Encoder

2.4.1.1. CNN Encoder

The process begins with Convolutional Neural Network (CNN), a VGG-style architecture featuring ten convolutional layers into four blocks seen in figure 3. Designed to extract high-level features from handwritten input. The model accepts a grayscale image with a height of 64 pixels and a dynamic width (B X 1 X 64 X W).Using consistent 3X3 convolutional kernels, the network identifies strokes and curves, progressively increasing the channel depth from 1 to 512 to capture complex handwriting patterns.

To stabilize training, the project utilizes residual Learning starting from the Conv3-Conv10 blocks and extending through the final convolutional stages. The residual blocks allow the original feature signal to bypass specific convolutional layers. This architecture mitigates the vanishing gradient problem, ensuring that the model remains stable as it reaches its maximum depth of 512 channels. It refines features across the blocks which the model uses 3X3 kernels for feature extraction while employing 1X1 convolutions to match dimensions whenever the channel count increases. It also prevents the model from overfitting to specific handwriting samples. Each residual block includes a 10% Drop Path rate, which randomly disables the residual path during training to force the model to learn more robust, generalized features. The layers use standard 2X2 max-pooling to down sample both height and width, the final blocks transition to height-Only Max-Pooling(2X1). This reduces the feature height to 4 pixels but leaves the width (w/4).Finally the

adaptive average pool collapses the training height to 1 pixel, and squeeze and permute operation reshapes the data into a sequence of [batch, sequence length, feature].
This transformation converts the 2D image into a 1D timeline with 512 dimensions per time step, perfectly formatting the data for the BiLSTM and attention stages.

$$Y_{i,j} = \sigma\left(\Sigma_{m=0}^{k-1}\Sigma_{n=0}^{k-1} W_{m,n} \cdot X_{i+m,j+n} + b\right) \quad \text{Equation 1}$$

where Yi, j is the output feature map at position (i,j), $W_{m,n}$ are the convolutional kernel weights (3×3), $X_{i+m,j+n}$ is the input region, b is the bias term, σ is the activation function ReLU, and k is the kernel size (3)

$$Y = F(X) + X \quad \text{Equation 2}$$

Where X is the input to the residual block, F(X) is the output of convolutional layers, and Y is the final output (input plus learned residual)

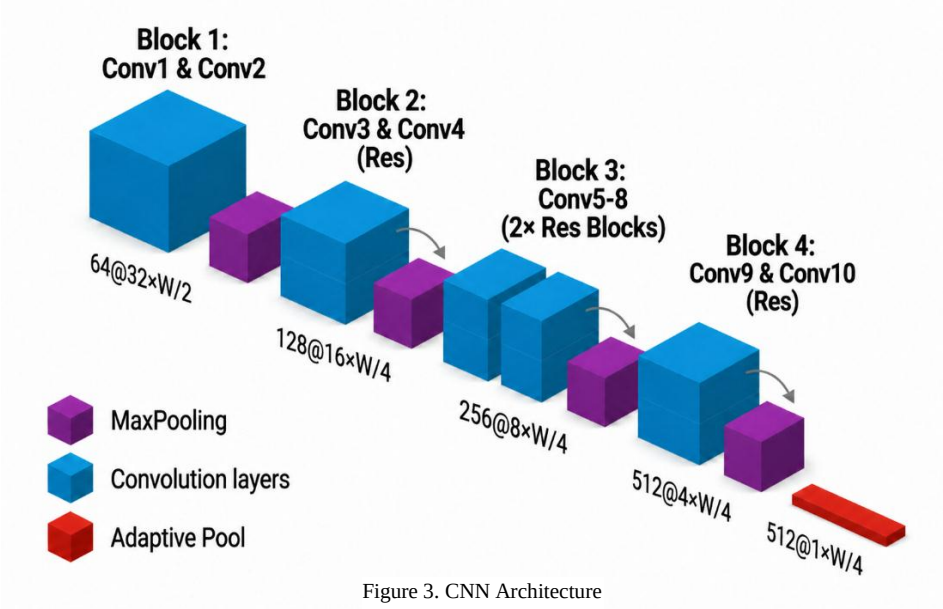

Figure 3. CNN Architecture

2.4.1.2. BiLSTM Encoder

The visual features are then compressed and passed to the bidirectional LSTM whose general architecture can be seen in figure 4. This stage acts as the bridge between pictures and language. By reading the feature sequence from both directions(left to right and right to left), the model gains a full understanding of the character order and the surrounding context. Prediction blocks. The sequence of 512-dimensional vectors produced by the CNN is fed into a Bi-LSTM network. It processes the feature sequence from both left to right and right to left simultaneously. A character often depends on the context of the strokes both preceding and following it, so by looking at the word from both ends, the model gains a deeper understanding of character connections and cursive.

The BiLSTM takes the 512-dimensional visual features and passes them through two hidden layers. The output from both directions is concatenated, resulting in an improved feature set that captures the full context of the word. The output of this stage is a sequence of vectors that represent the “hidden state “ of the word at every horizontal position, which is then passed to the Bahdanau Attention Mechanism for final

decoding. A dropout of 0.3 is applied between the BiLSTM layers. Ensuring that the model learns generalized linguistic patterns rather than the specific training example which prevents the model from becoming too rigid or “memorizing” the training sequence

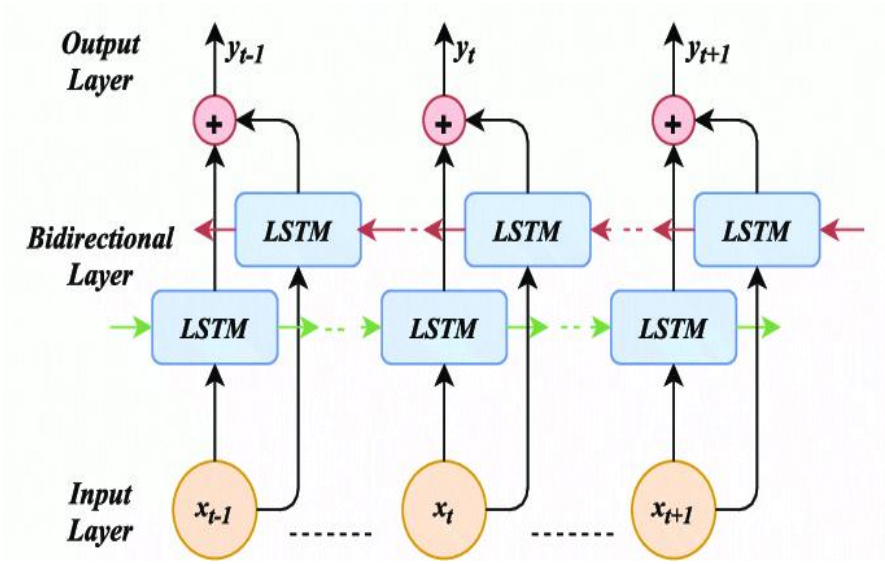

Figure 4. BiLSTM Architecture

2.4.1.2.1. LSTM Cell Gating

Each step of the sequence is governed by internal gates that decide what information to keep or discard.

Forget Gate : $f_t = \sigma(W_f \cdot x_t + U_f \cdot h_{t-1} + b_f)$ Equation 3

Input Gate : $i_t = \sigma(W_i \cdot x_t + U_i \cdot h_{t-1} + b_i)$ Equation 4

Output Gate : $o_t = \sigma(W_o \cdot x_t + U_o \cdot h_{t-1} + b_o)$ Equation 5

Candidate Cell State : $\tilde{C}_t = tanh(W_c \cdot x_t + U_c \cdot h_{t-1} + b_c)$ Equation 6

Cell State Update : $C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$ Equation 7

Hidden State Update : $h_t = o_t \odot tanh(C_t)$ Equation 8

2.4.1.2.2. Bidirectional Concatenation

The "Bidirectional" aspect is mathematically represented by combining the forward and backward and hidden state.

$$h_t = concat(\overrightarrow{h_t}, \overleftarrow{h_t}) \quad \text{Equation 9}$$

This concatenation produces a 512-dimensional output (256 from the forward direction and 256 from the backward direction), forming a context-aware representation that captures both the preceding and following stroke context at every position in the sequence.

2.4.1.2.3. Regularization(Dropout)

To ensure the research is robust against different handwriting speeds and styles, dropout is applied to the hidden states:

$$H_t(regularized) = Dropout(H_t, p) \quad \text{Equation 10}$$

This forces the network to find multiple "paths" to identify a character, leading to better accuracy on messy or unfamiliar handwriting.

2.4.2.Prediction Block

2.4.2.1.Bahdanau Attention

It acts as an attentional shift that calculates specific weight for the encoder's features. It is used to create a context vector at each time step, ensuring the model focuses only on the relevant pen stroke for the character currently being predicted. The architecture of bahdanau attention is shown below in figure 5.

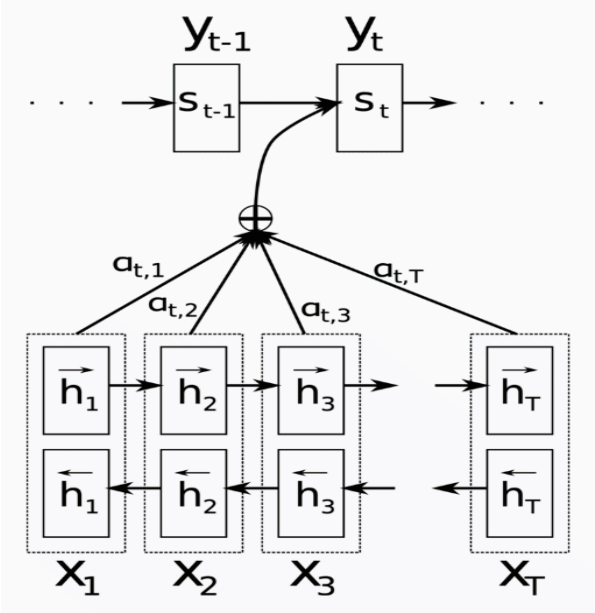

Figure 5. Bahdanau Attention

The Bahdanau Attention is implemented to dynamically focus on relevant parts of the encoder output at each decoding step. Attention score for each encoder position j is computed as:

$$E_{proj} = W_e \cdot h_{enc} \quad \text{Equation 11}$$

Where E_{proj} is the projected encoder features, W_e is the encoder projection weight matrix, and h_{enc} is the encoder output (hidden states from the BiLSTM).

$$D_proj = W_d \cdot s_t \quad \text{Equation 12}$$

Where D_{proj} is the projected decoder feature, W_d is decoder projection weight matrix, S_t is decoder hidden state at time step t

$$e_ij = v_a^T \cdot tanh\left(W_a \cdot s_{i-1} + U_a \cdot h_j\right) \quad \text{Equation 13}$$

Where e_{ij} is alignment score between decoder state and encoder state, v learned energy weight vector, tanh is hyperbolic tangent activation, W_a is the weight matrix, si-1 is the decoder hidden state at previous time stamp, U_a is the weight matrix that projects the encoder hidden state to the attention space, h_j is the encoder hidden state at position j

$$\alpha_{ij} = exp\left(e_{ij}\right)/\Sigma_{k=1}^{T_x} exp(e_{ik}) \quad \text{Equation 14}$$

Where a_{ij} is the normalized attention weight,

$$c_i = \Sigma_{j=1}^{T_x} \alpha_{ij} \cdot h_j \quad \text{Equation 15}$$

where c_i is the context vector at decoder time step

$$s_i = f(s_{i-1}, y_{i-1}, c_i) \quad \text{Equation 16}$$

where s_i is the updated decoder hidden state at step i, f represents the LSTM cell function that computes the new hidden state.

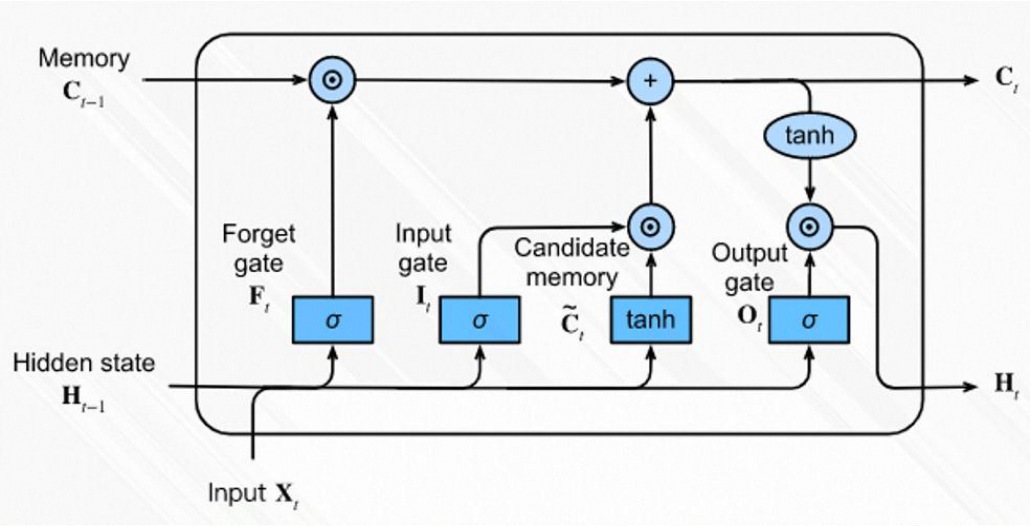

Figure 6. LSTM Architecture

2.4.2.2. LSTM Decoder and SoftMax Layers

This layer handles the sequential generation and final character selection. The LSTM shown in figure 6 is used to maintain the linguistic flow of the word. A unidirectional LSTM responsible for generating the output character sequence one token at a time, guided by the attention mechanism. At each time step it receives 3 inputs concatenated together: a 64-dimensional embedding for the previously predicted character and a 512-dimensional context vector from the Bahdanau Attention module. This concatenated 576-dimensional vector is fed into the LSTM cell. The unidirectional LSTM decoder utilizes the identical gating mechanisms defined in Equations 3 through 8. The key architectural difference lies in the input vector: whereas the encoder LSTM receives the visual feature sequence Xt from the CNN, the decoder LSTM receives the concatenated [64 + 512 = 576] dimensional vectors comprising the previous character embedding and the Bahdanau context vector at each decoding step. During training, the source of the previous character input is governed by an adaptive teacher forcing schedule: the ratio begins at 1.0, where ground-truth characters are always fed as decoder inputs, and decays gradually over epochs so the model increasingly relies on its own predictions. This progressive shift bridges the gap between training and inference behavior, improving generalization to unseen handwriting styles.

The final step is to transform the hidden representation into a probability distribution over the character vocabulary. The character vocabulary comprises 26 lowercase letters, 26 uppercase letters, and 10 digits (0–9), supplemented by a space token and an end-of-sequence token, yielding a total vocabulary size of 64. It is achieved through a linear projection followed by a SoftMax activation function. The linear projection maps the decoder hidden state to a 64-dimensional logit vector, which is then normalized by SoftMax to produce a probability distribution over the 64-character vocabulary. It is defined as:

$$softmax(z_i) = exp(z_i)/\Sigma_{j=1}^{n} exp(z_j) \qquad \text{Equation 17}$$

Where Z_i is the input value at position i, exp is the exponential function, n is the total number of elements in the input vector, and the vector contains one probability value for every character in the vocabulary, and all values sum to 1.The model then selects the character with the highest probability as the prediction for time step.

3. Experimental Overview

The IAM handwritten dataset from the FKI database is used for the experiment. Preprocessing is carried out using Python programming to make the data clean and ready for processing. For this, loading the images as grayscale, Normalizing the image to [0,1], resizing the image(height = 64, preserving aspect ratio). An Attention-based Seq2Seq model is used for the model training. At the conclusion of the experiment Accuracy, CER, WER are assessed using testing data to find how the model performs in real world scenario.

3.1. Result and Analysis

3.1.1. Training Process

The model was trained for 50 epochs on the IAM handwriting dataset. Figure 7 shows the training and validation metrics across all epochs. Loss Curves (Top Left):Training loss decreased rapidly from 2.50 to approximately 0.80,showing strong learning capability. Validation loss decreased from 1.95 to 0.95 and

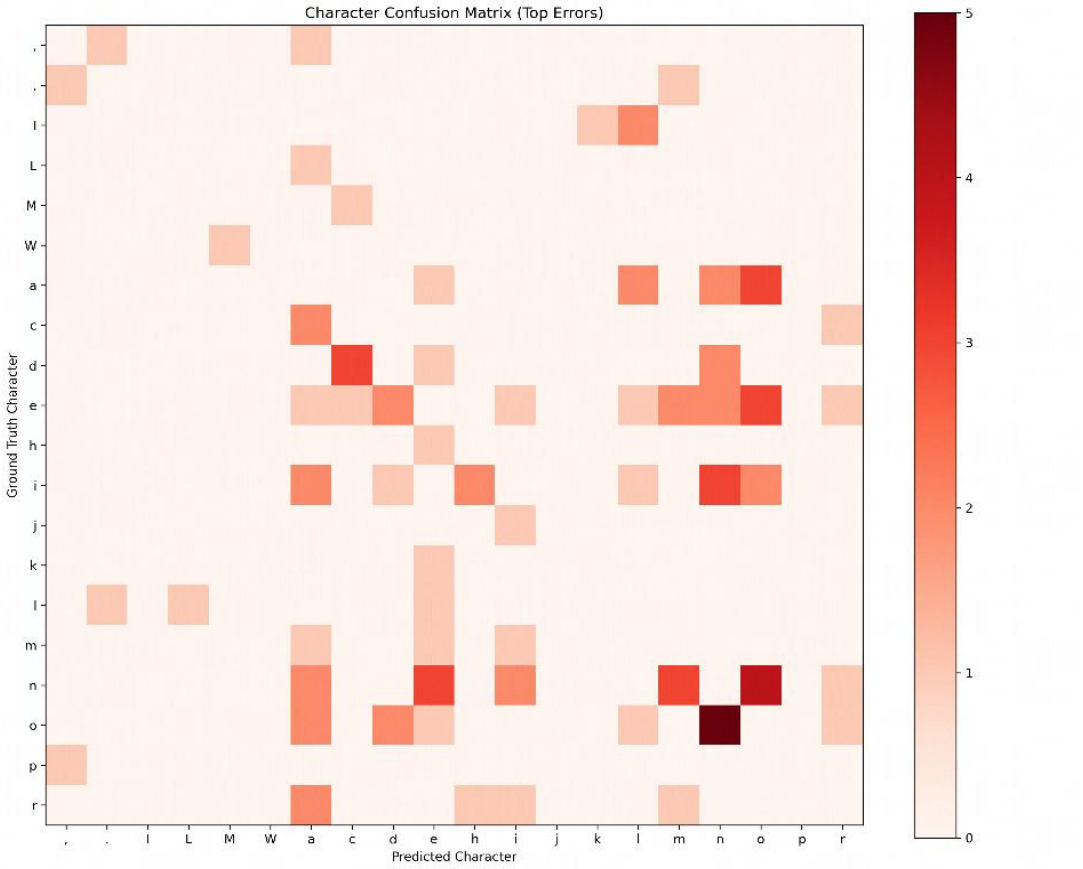

Figure 7. Confusion Matrix

stabilized around epoch 10. Both curves converge and stabilize after epoch 15 and the small gap between train and validation loss indicates minimal overfitting.

- Word Error Rate - WER (Top Right): The figure shows that the model learned quickly for the first 20 epochs and stabilized after 30 epochs. The model achieved the validation WER of 15.78% which shows that the model is correctly predicting approximately 84.22% of words. The stable WER suggests that the model converges without degradation.
- Character Error Rate-CER(Bottom Left): The smooth decline in CER from 46% to 6% suggests that the attention mechanism effectively learned to focus on relevant image regions for each character. Lower CER compared to WER indicated that when the model makes errors, they are typically single-character mistakes rather than the completely wrong words. It can also be seen that most improvement occurred in the first 15 epochs.

- Learning Rate schedule(Bottom Right): The smooth curve with decreasing learning rate indicates the proper implementation of the scheduling techniques and also shows that oscillations were prevented and helped the model settle into optimal weights.

3.1.1.1. Confusion Matrix

The confusion matrix reveals the most common character misclassification, with the darkest cells indicating frequent errors. Notable confusion includes ‘o’ to ‘n’, ‘o’ to ‘a’, ‘e’ to ‘c’, ‘i’ to ‘l’, and ‘n’ to ‘m’, which are visually similar characters in cursive handwriting recognition and account for the majority of the 4.28%-character error rate.

The attention visualization confirms that the Bahdanau attention mechanism successfully learned meaningful alignment between the input image and output characters. These patterns indicate that the attention mechanism is not randomly guessing but has learned to locate and decode the characters sequentially from the handwritten images.

3.1.1.2. Attention Mask

This figure shown below is the attention visualization of the model on four sample handwritten word images. For the word "Gaitskel", the attention map displays a clear diagonal pattern, confirming that the model attends to image regions sequentially from left to right, character by character. For shorter words like "to" and "a", the attention is concentrated in a narrow vertical band since the actual handwritten content occupies only a small portion of the padded image width. The word "chael" shows scattered but progressive attention across the relevant image region. Overall, these heatmaps validate that the Bahdanau attention mechanism has successfully learned meaningful spatial-sequential alignment between input image positions and output characters, rather than attending randomly across the image.

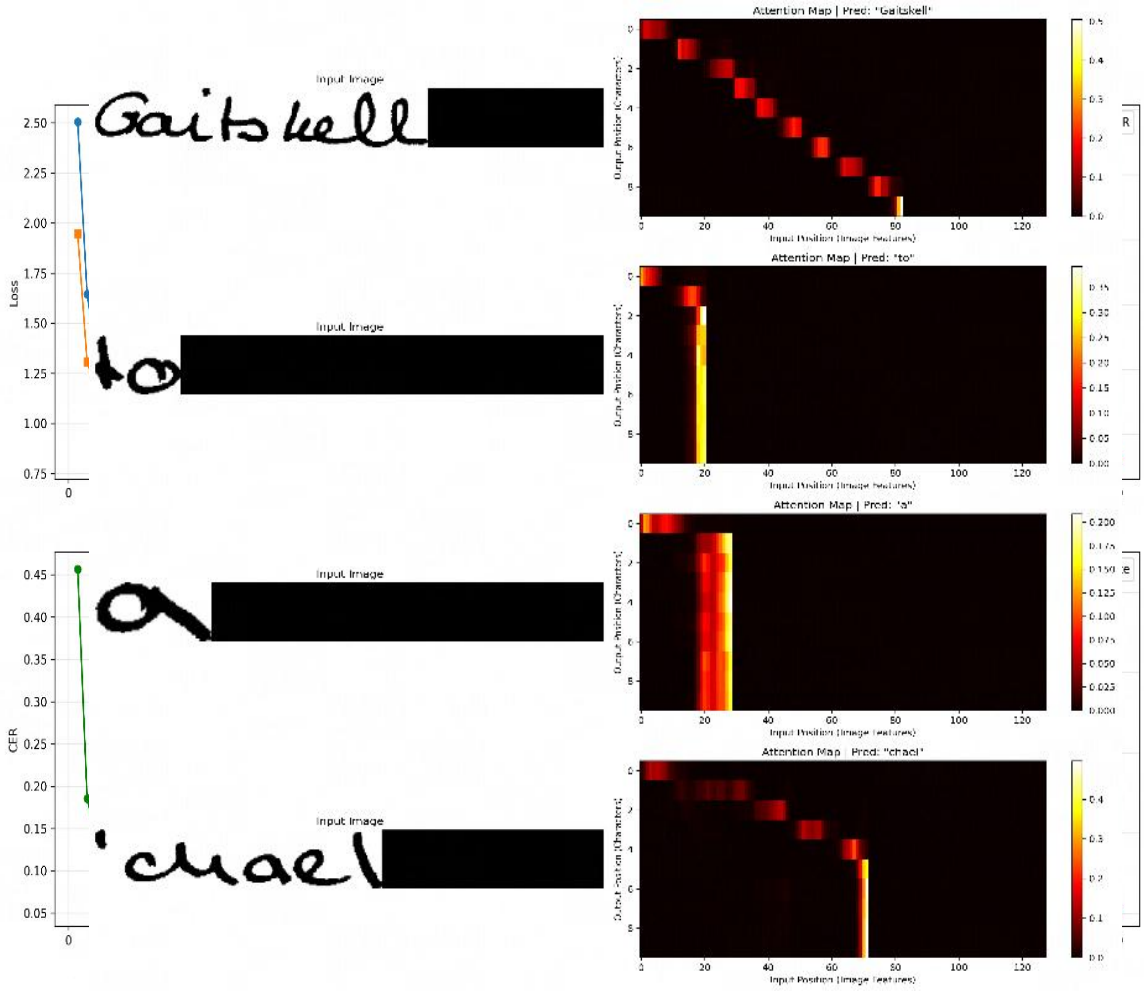

Figure 8 Attention Visualization

3.1.2. Test Set Performance

To evaluate true generalization, the official IAM test set was strictly held out from the beginning of training and was not used for any hyperparameter tuning or model selection. The model achieved a final test WER of 13.59% and a CER of 4.28%. Comparing this to the validation WER of 15.78% (a difference of 2.19%), the test performance shows a noticeable improvement. This discrepancy is expected and directly attributable to our training strategy: while the 15.78% metric represents the standard model's validation performance at a given epoch, the final test metrics were evaluated using the Stochastic Weight Averaging (SWA) and Exponential Moving Average (EMA) checkpoints. SWA averages the model weights over the final training epochs, which acts as a powerful regularizer and consistently yields a model that generalizes better than the single best validation checkpoint. Furthermore, the IAM dataset's inherent variability means the specific writers in the official validation split naturally possess slightly more challenging handwriting styles than those in the test split. These results confirm that the model generalized robustly to completely unseen data.

4. Model Comparison

Table 1 compares the proposed model against recent HTR methods evaluated on the IAM handwriting dataset. The proposed model achieves 13.59% WER and 4.28% CER, outperforming all compared methods across available metrics. Compared to Kass et al. (2022), which employs a similar attention-based encoder-decoder architecture with ResNet and BiLSTM, the proposed model achieves a 1.81% reduction in WER and 2.22% reduction in CER. Against Shrestha et al. (2023) and Jain et al. (2021), the proposed model reduces CER by 5.05% and 4.61% respectively.

Table 1. Model Comparison

Model	**WER**	**CER**
Proposed Model	13.59%	4.28%
Kass et al.(2022)	15.40%	6.50%
Shrestha et al.(2023)	N/A	9.33%
Jain et al.(2021)	N/A	8.89%

These results demonstrate that the proposed architecture, enhanced through systematic regularization comprising DropPath, EMA, SWA, and adaptive teacher forcing, achieves state-of-the-art performance among CRNN-based frameworks on the IAM dataset.

5. Discussion & Conclusion

This paper presented a hybrid CRNN architecture integrating a Seq2Seq Bahdanau attention mechanism for handwritten text recognition on the IAM handwriting dataset. The proposed model achieved 13.59% WER and 4.28% CER, outperforming recent comparable methods including Kass et al. (2022), Shrestha et al. (2023), and Jain et al. (2021). The attention mechanism's ability to dynamically focus on relevant spatial regions during character decoding was validated through attention map visualizations, which revealed meaningful diagonal alignment patterns corresponding to sequential character progression.

With 86.41% word-level accuracy and 95.72% character-level accuracy on the test set, the model demonstrated strong generalization to unseen handwriting styles. These results validate that systematic regularization combining dropout, weight decay, DropPath, EMA, SWA, and adaptive teacher forcing decay can yield robust HTR performance without reliance on synthetic data, external lexicons, or transformer pre-training. While the Seq2Seq architecture incurs higher inference latency than CTC-based decoding, the substantial accuracy gains make it a practical choice for applications where recognition quality is prioritized over inference speed.

6. Future Enhancement

Future work will explore extending the model beyond isolated words to recognize full lines and paragraphs. Testing on diverse datasets such as RIMES and CVL will further validate the model's generalization across different languages and writing styles. Replacing the Bahdanau attention mechanism with a Transformer-based decoder is expected to improve both speed and accuracy. Finally, applying knowledge distillation techniques will help compress the model for faster, real-time deployment on resource-constrained devices.

Acknowledgement

We would like to express our sincere gratitude to our supervisor, Er. Nirajan Acharya, for his valuable guidance, encouragement, and continuous support throughout the course of this research. His insights and feedback have been instrumental in shaping the direction and quality of our work.

We also affirm that all authors have contributed equally to the research and preparation of this paper.

References

[1] D. Kass and E. Vats, "AttentionHTR: Handwritten Text Recognition Based on Attention Encoder-Decoder Networks," in Proc. Int. Workshop Document Analysis Systems (DAS), 2022.

[2] A. A. A. Ali and S. Mallaiah, "Intelligent handwritten recognition using hybrid CNN architectures based-SVM classifier with dropout," Journal of King Saud University - Computer and Information Sciences, vol. 34, pp. 3294-3300, 2022.

[3] A. A. A. Alshawi, J. Tanha, and M. A. Balafar, "An Attention-Based Convolutional Recurrent Neural Networks for Scene Text Recognition," IEEE Access, vol. 12, pp. 8123-8134, 2024.

[4] T. Jain, R. Sharma, and R. Malhotra, "Handwriting Recognition for Medical Prescriptions using a CNN-Bi-LSTM Model," in Proc. 6th Int. Conf. for Convergence in Technology (I2CT), 2021.

[5] J. Michael, R. Labahn, T. Grüning, and J. Zöllner, "Evaluating Sequence-to-Sequence Models for Handwritten Text Recognition," in Proc. Int. Conf. on Document Analysis and Recognition (ICDAR), 2019, pp. 1286-1293.

[6] G. Retsinas, K. Nikolaidou, and G. Sfikas, "Enhancing CRNN HTR Architectures with Transformer Blocks," in Proc. 18th Int. Conf. on Document Analysis and Recognition (ICDAR), 2024, pp. 425-440.

[7] R. Shrestha, O. Shrestha, M. Shakya, U. Bajracharya, and S. Panday, "Offline Handwritten Text Extraction and Recognition Using CNN-BLSTM-CTC Network," International Journal on Engineering Technology (InJET), vol. 1, no. 1, pp. 166-180, Nov. 2023.

[8] S. K. Sunori and S. Sumithra, "Enhancing Handwritten Text Identification through a Hybrid CNN-RNN Method," in Proc. 3rd Int. Conf. on Optimization Techniques in the Field of Engineering (ICOFE), 2025.

[9] N. Teslya and S. Mohammed, "Deep Learning for Handwriting Text Recognition: Existing Approaches and Challenges," in Proc. 31st Conf. of Open Innovations Association (FRUCT), Apr. 2022.

[10] S. Yadav, S. Singh, and P. K. Shukla, "Handwritten Medical Text Recognition Using Region-Based CRNN and Connectionist Temporal Classification," World Journal of Pharmaceutical Research, vol. 14, no. 15, pp. 1461-1486, 2025.

[11] L. Kumari, S. Singh, V. V. S. Rathore, and A. Sharma, "Lexicon and Attention based Handwritten Text Recognition System," Machine Graphics & Vision, vol. 31, no. 1/4, pp. 75-92, 2022, arXiv:2209.04817.

[12] P. T. T. Truc, D. H. Nam, H. T. D. Khoa, and V. N. L. Duy, "HTR-ConvText: Leveraging Convolution and Textual Information for Handwritten Text Recognition," 2025.