Techno-Science Research Journal V14 (Special Issue) (2026) P 60-65

Content list available at ITC

Techno-Science Research Journal

Journal Homepage: http://techno-srj.itc.edu.kh/

Techno-Science Research Journal

Recognizing Khmer Handwritten Digits with the Power of Sequential RNNs

Kimlong Ngin [1*], Dona Valy [1], Kimhor Phoeurn[2] , Seyha Va[2], Bunkhloem Pel[2], Kana Ty[2], Sambat Yin[2], Rotha Pho[2]

[1] *Research and Innovation Center, Institute of Technology of Cambodia, Russian Federation Blvd., P.O. Box 86, Phnom Penh, Cambodia*

[2] *Department of Applied Mathematics and Statistics, Institute of Technology of Cambodia, Russian Federation Blvd., P.O. Box 86, Phnom Penh, Cambodia*

Received: 20 August 2025; Revised: 28 January 2026; Accepted: 05 February 2026; Available online: 31 July 2026

Abstract: *Recognizing handwritten digits is a fundamental aspect of optical character recognition (OCR), with broad applications in areas such as digital archiving, data entry, and assistive technologies. Khmer digits present distinctive challenges due to their intricate shapes and high variability in individual handwriting styles, making the development of accurate recognition systems particularly demanding. Unlike conventional approaches that primarily rely on image-based inputs, this study adopts a sequential framework using coordinate data, which captures the temporal dynamics of pen strokes and preserves the natural writing sequence. This aspect is especially critical for Khmer digits, where stroke order follows well-defined structural rules. Three recurrent neural network architectures such as Long Short-Term Memory (LSTM), Bidirectional LSTM (Bi-LSTM), and Gated Recurrent Unit (GRU) were evaluated, with data augmentation applied to improve robustness. A custom dataset of 1,583 handwritten sequences was expanded to 12,337 samples through rotation-based augmentation, partitioned into 60% training, 20% validation, and 20% testing. The experimental evaluation reported test accuracies of 95.27% for LSTM, 94.95% for Bi-LSTM, and 95.58% for GRU, with GRU showing the most promising results. These outcomes validate the effectiveness of sequential modeling for Khmer digit recognition and emphasize the value of RNN-based methods for complex handwriting systems.*

Keywords: *Khmer Handwriting; Khmer Digit Recognition; RNN Sequential data; Points (x, y)*

1. INTRODUCTION

Handwritten Khmer digit recognition is a fundamental challenge in machine learning and computer vision, with significant applications across education, business, and government sectors. In education, such technology can be deployed in mobile applications that help children practice writing digits by providing real-time feedback on stroke order, direction, and correctness. Online handwriting recognition also enables interactive systems such as digital examination tools, e-learning platforms, and secure authentication systems where users input digits via handwriting. Efficient recognition is therefore essential for converting handwritten input into digital formats in real time, ensuring both accuracy and usability across different domains. Over the years, numerous approaches have been proposed for handwritten digit recognition ranging from traditional machine learning methods such as Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) **[1]** , to deep learning approach such as leveraging Convolutional Neural Networks (CNN) **[2]** and Recurrent Neural Network (RNN) **[3]** . Specifically, these deep learning approaches have demonstrated extraordinary performance in handwritten recognition systems. This paper proposes shifting from traditional image-based inputs to a sequential data approach, modeling the order and dynamics of strokes as they are drawn. By capturing the temporal information inherent in pen movement, this method better reflects the unique characteristics of Khmer handwriting and improves recognition accuracy.

* Corresponding author: Kimlong Ngin
E-mail: ngin.kimlong@itc.edu.kh

2. RELATED WORKS

Khmer handwriting recognition has attracted interest due to its role in digitizing Khmer script. While several studies have investigated Khmer text recognition, specific research handwritten Khmer digit recognition is limited. This section reviews key contributions in Khmer handwritten text recognition and highlights the gap our work aims to fill. Annanurov et al. [4] introduced a lightweight deep learning architecture, 2+1CNN, for offline Khmer handwritten text recognition. The model comprises two convolutional layers and a fully connected layer. It achieved an average recognition accuracy of 94.9%. Designed for resource-constrained environments, their system adopted a one-against-all classification approach by training 33 distinct binary classifiers for Khmer consonants. The results demonstrated that compact CNNs can effectively process complex scripts like Khmer. Another relevant study by Annanurov et al. [5] applied Convolutional Neural Networks (CNNs) for recognizing handwritten Khmer syllables. Handwritten samples were preprocessed through grayscale conversion, size normalization to 32×32 pixels, and binary thresholding. A similar one-against-all training strategy was used with an average accuracy of 94.85%. In contrast to deep learning, Meng et al. [6] proposed a hybrid neural architecture combining Self-Organizing Maps (SOM) and Multilayer Perceptrons (MLP) for Khmer character recognition. Input images (20×20 pixels) were binarized and vectorized into 400-dimensional vectors. SOM was used to cluster inputs, followed by MLP-based classification for each cluster. Although limited by a small dataset of 135 samples derived from printed fonts, this system showcased an effective hierarchical classification pipeline. The authors found that a training loss of 10^{-29} significantly improved performance, increasing the recognition rate from 72% to 94.13% on a 10% noise condition of test set. However, the model showed a drop in accuracy for 65% noised train set and 30% accuracy for noised unseen data. In addition to research on Khmer digits, international studies on handwritten digit recognition provide important benchmarks. For instance, Ullah at al. [7] evaluated various models on the MNIST dataset, which contains 70,000 28×28 grayscale images of digits. Traditional models such as SVM, k-NN, and Random Forest were tested alongside deep learning models including CNNs, RNNs, and LSTMs. Hybrid approaches, notably CNN-SVM and CNN-LSTM, were also explored, combining spatial feature extraction with sequential modeling. The study found that the CNN-SVM hybrid achieved the highest accuracy of 99.30%, outperforming standalone CNNs (98.96%) and other models. While MNIST digits are simpler and more standardized than Khmer digits, these benchmarks provide useful reference points for deep learning performance. Khmer handwritten digits present additional challenges, including complex stroke patterns, higher writing variability, and limited dataset size, which make offline image-based recognition less straightforward. Sequential models such as LSTM, Bi-LSTM, and GRU are particularly suited for online handwriting recognition, as they can leverage stroke order and temporal dynamics information not available in static images. This makes sequential approaches more appropriate for real-time applications, which is the focus of our study.

3. METHODOLOGY

3.1 Data Collection

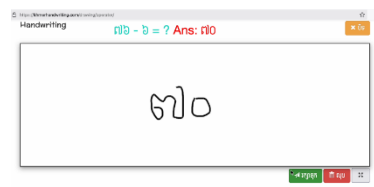

Fig. 1. Web-based Interface used for Collecting Online Handwritten Khmer Digit Sequences

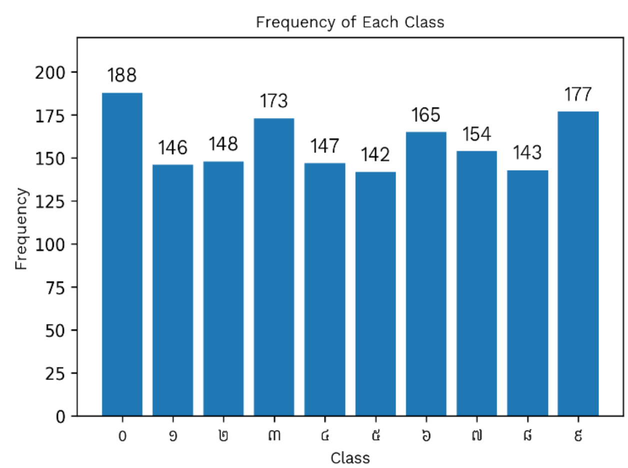

Fig. 2. Distribution of Collected Khmer Digit Samples by Class

To construct a dataset of Khmer handwritten numbers, a web-based platform as shown in Fig. 1 was developed to allow users to draw numbers using their mouse or touchscreen. Each entry includes a digit label, and a stroke trajectory saved in a comma-separated format: $x_1, y_1, x_2, y_2, \ldots$. Multiple strokes are separated by a delimiter (#). A total of 1,583 samples were collected from Cambodian university students with ages from 18 to 22. Both male and female participants contributed to the data collection reflect the handwriting styles typical among young adults in Cambodia.

Fig. 2 illustrates the class distribution ranged from 142 to 188 samples per digit, with digit 0 being most frequent and digit 5 the least. While there is some class imbalance, it is relatively small and does not heavily skew the dataset.

3.2 Data Preprocessing

The raw data underwent a multi-stage preprocessing pipeline to prepare it for the models. The pipeline began with cleaning the dataset by removing any entries with missing values. Each digit was then segmented into individual strokes using the '#' delimiter. To ensure uniformity across different drawing scales and positions, all stroke coordinates were normalized to a [0,1] range using the min-max scaling formulas shown in **(Eq. 1)** and **(Eq. 2)**.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad \text{(Eq. 1)}$$

$$y_{scaled} = \frac{y - y_{min}}{y_{max} - y_{min}} \quad \text{(Eq. 2)}$$

Where x and y are the original coordinates, while $x_{min}, x_{max}, y_{min},$ and y_{max} represent the minimum and maximum coordinate values for that stroke. Subsequently, each main stroke was divided into uniform substrokes to create fixed-size input sequences for the models. Each substroke was set to contain a fixed length of 8 coordinate pairs. The total number of substrokes for a given main stroke was determined by (Eq. 3).

$$\#\ \text{substrokes} = \frac{\text{total points of main stroke}}{\#\ \text{segments}} \quad \text{(Eq. 3)}$$

Where the number of segments per substroke is 8. If the final substroke contained fewer points than required, it was padded with $(0, 0)$ to ensure a uniform size across all inputs.

3.3 Data Preparation and Augmentation

This dataset was split into a 60% training set, 20% validation set, and a 20% test set. To further enhance dataset diversity and improve generalization, a rotation-based data augmentation technique was applied on the train set. The original stroke coordinates were rotated using angles from -30° to 30°, in 5° increments. This simulates natural variations in handwriting while preserving the character's structure. Mathematically, each point (x, y) in a stroke is rotated about the origin using a standard two-dimensional rotation matrix:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Resulting in:

$$x' = x\cos(\theta) - y\sin(\theta) \quad \text{(Eq. 4)}$$

$$y' = x\sin(\theta) + y\cos(\theta) \quad \text{(Eq. 5)}$$

where θ is the rotation angle in radians. Each stroke with (x, y) coordinates was transformed for every $\theta \in \{-30°, -25°, \ldots, 30°\}$. As a result, each original sample was augmented with multiple rotated versions which produce a more robust and representative training set.

3.4 Model Architecture

To model the temporal dynamics inherent in handwritten stroke data, three Recurrent Neural Network (RNN) architectures were evaluated. RNNs are specifically designed for sequential data, making them ideal for interpreting the patterns of how a digit is drawn over time.

3.4.1 Long Short-Term Memory (LSTM)

The primary architecture explored was the Long Short-Term Memory (LSTM) network **[8]**, a variant of RNN. LSTMs are explicitly designed to overcome the vanishing and exploding gradient problems that can affect simpler RNNs, allowing them to effectively learn long-range dependencies within the pen stroke sequences.

An LSTM cell operates as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1} x_t] + b_f) \quad \text{(Eq. 6)}$$

$$i_t = \sigma(W_i \cdot [h_{t-1} x_t] + b_i) \quad \text{(Eq. 7)}$$

$$\tilde{c_t} = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad \text{(Eq. 8)}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad \text{(Eq. 9)}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad \text{(Eq. 10)}$$

$$h_t = o_t \odot \tanh(c_t) \quad \text{(Eq. 11)}$$

Here, σ is the sigmoid function, $\odot$ is the element-wise multiplication and x_t is the input at time t . The cell state c_t acts as a memory to carry long-term information.

3.4.2 Bidirectional-LSTM (Bi-LSTM)

To capture contextual information from the entire stroke, a Bidirectional LSTM (Bi-LSTM) **[9]** was also implemented. Unlike a standard LSTM which only processes the sequence in a forward direction, the Bi-LSTM utilizes two separate LSTM layers: one processing the sequence from start to finish, and the other from finish to start. It consists of two LSTMs: one reads the input from $t = 1$ to T, and the other from $t = T$ to 1. The outputs from both directions are concatenated:

3.4.3 Gated Recurrent Unit (GRU)

The Gated Recurrent Unit (GRU) **[10]**, a more modern and computationally efficient variant of the LSTM, was also evaluated. The GRU simplifies the LSTM architecture by combining the forget and input gates into a single update gate and merging the cell state and hidden state. This streamlined design often allows for faster training without a significant loss in performance, making it a strong candidate for this recognition task.

The GRU equations are:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \quad \text{(Eq. 15)}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \quad \text{(Eq. 16)}$$

$$\widetilde{h_t} = \tanh(W_h \cdot [r_t \odot h_{t-1}, x_t] + b_r) \quad \text{(Eq. 17)}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widetilde{h_t} \quad \text{(Eq. 18)}$$

3.5 Model Training

A set of regularization and optimization techniques was utilized to increase the robustness and generalization capabilities of the models. A dropout rate of 0.3 was applied between the LSTM/GRU layers and fully connected layers to avoid overfitting. The Adam optimizer was used for optimization. Additionally, CrossEntropyLoss function was utilized to handle classification tasks. In addition to these steps, hyperparameters tuning was also done using grid search to find the best hyperparameters. It includes going through various learning rates from 0.01, 0.001, 0.0001 with the number of epochs of 30, 40, and 50 to ensure the final model obtains the best performance. The details of the explored hyperparameters are summarized in **Table 1**.

Table 1: Randomized Hyperparameter Tuning

Hyperparameter	Search Space
Hidden Size	256
Number of Layers	4
Dropout Rate	0.3
Batch Size	32
Number of Epochs	[30, 40, 50]
Learning Rate	[0.1, 0.01, 0.001, 0.0001]

$$\overrightarrow{h_t} = LSTM_{fwd}(x_t) \quad \text{(Eq. 12)}$$

$$\overleftarrow{h_t} = LSTM_{bwd}(x_t) \quad \text{(Eq. 13)}$$

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}] \quad \text{(Eq. 14)}$$

3.6 Model Evaluation

To evaluate the effectiveness of different recurrent neural network architectures in classifying the stroke point of written Khmer digits, two key evaluation metrics were assessed: Accuracy and F1 score. Since the F1 score is derived from Precision and Recall, these two metrics are also defined below to provide clarity.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad \text{(Eq. 19)}$$

$$Precision = \frac{TP}{TP + FP} \quad \text{(Eq. 20)}$$

$$Recall = \frac{TP}{TP + FN} \quad \text{(Eq. 21)}$$

$$F1\,Score = 2\left(\frac{Precision \times Recall}{Precision + Recall}\right) \qquad \text{(Eq. 22)}$$

where TP: True Positive, FP: False Positive, FN: False Negative

4. RESULTS

As shown in **Table 2**, all models demonstrate strong performance, with accuracies exceeding 95% which indicates their effectiveness in capturing sequential patterns in the data.

Table 2: Model Performance Comparison of Test Set

Model	Accuracy (Train/Test)	F1 Score	LR	Epochs
LSTM	0.9938/0.9527	0.9530	0.001	30
Bi-LSTM	0.9991/0.9495	0.9505	0.0001	40
GRU	0.9973/0.9558	0.9561	0.001	50

The LSTM model achieved a training accuracy of 99.38% and a test accuracy of 95.27% and an F1 score 95.30%, establishing a reliable baseline for sequential classification. The slight drop reflects mild overfitting, which is expected given the small original dataset of 1,583 samples, even after augmentation to 12,337 samples. The Bi-LSTM model, despite its bidirectional structure, reached a training accuracy of 99.91% and a test accuracy of 94.95% and an F1 score of 95.05% which suggests that added complexity may introduce redundant representations and minor overfitting. The GRU model performed best, with training accuracy 99.73%, test accuracy 95.58%, and F1 score of 95.61%. Its marginal superiority is due to a simpler gating mechanism, combining input and forget gates into a single update gate and using a reset gate, which reduces parameters, stabilizes gradients, and efficiently captures stroke-level temporal patterns in Khmer handwritten digits. Overall, all models generalize reasonably well, with GRU striking the best balance between complexity and performance.

While the overall results are strong, a deeper inspection of the confusion matrices **Fig. 3** reveals patterns in model misclassifications. Across all models, most errors occur between visually similar digits. For example, digit 5 is frequently misclassified as 4, while digit 4 is often mistaken for 5 or 9. The LSTM model shows 15 total misclassifications, the Bi-LSTM model 16, and the GRU model only 10, confirming its slightly stronger generalization.

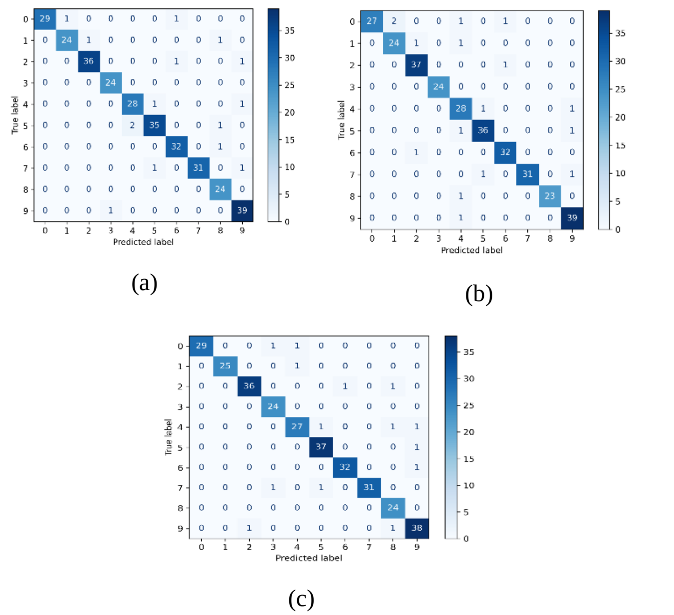

Fig. 3. Test set confusion matrices illustrating classification results for (a) LSTM, (b) Bi-LSTM, and (c) GRU architectures.

5. CONCLUSION AND DISCUSSION

This research examined the role of recurrent neural networks in recognizing Khmer handwritten digits, addressing a gap in previous studies that have primarily concentrated on CNN-based image recognition. Through an evaluation of LSTM, Bi-LSTM, and GRU models, we showed that sequential architectures are capable of capturing stroke order and temporal dependencies, offering advantages over static image-based approaches. Among the models tested, the GRU achieved the strongest results, with 95.58% accuracy and an F1 score of 95.61%, demonstrating its effectiveness in handling handwriting variability.

Despite these encouraging outcomes, several constraints remain. The dataset used was relatively limited in size and drawn from a narrow demographic, potentially affecting generalizability. Moreover, the augmentation process relied solely on angular variations, without incorporating other forms of transformation such as scaling, translation, elastic distortion, or noise injection. Including these methods in future work could enrich variability and further strengthen model generalization.

Additionally, this study primarily focused on stroke-based sequential modeling without a direct comparison to image-based CNNs. Prior research on Khmer handwritten text recognition using CNNs generally reports accuracies of 94–95%, which is on par with the performance achieved by our GRU model. This suggests that RNN-based methods provide a viable alternative, particularly when temporal stroke data is available. Future work should also consider hybrid CNN–RNN architectures to exploit both spatial and temporal features for improved recognition. In conclusion, the findings underscore the potential of recurrent neural networks as complementary to CNN-based methods and establish a foundation for practical applications, such as mobile educational platforms and interactive systems for teaching and practicing Khmer numerals.

ACKNOWLEDGEMENTS

The authors would like to express their sincere gratitude to the ViLa Lab for their generous support, insightful feedback, and access to essential resources. Their contributions have played a vital role in enhancing the depth and quality of this research.

REFERENCES

[1] C. Kaensar, "A Comparative Study on Handwriting Digit Recognition Classifier Using Neural Network, Support Vector Machine and K-Nearest Neighbor," in *The 9th International Conference on Computing and InformationTechnology (IC2IT2013)*, P. Meesad, H. Unger, and S. Boonkrong, Eds., Berlin, Heidelberg: Springer, 2013, pp. 155–163. doi: 10.1007/978-3-642-37371-8_19.

[2] S. Ahlawat, A. Choudhary, A. Nayyar, S. Singh, and B. Yoon, "Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN)," *Sensors*, vol. 20, no. 12, Art. no. 12, Jan. 2020, doi: 10.3390/s20123344.

[3] I. J. L. Paul, S. Sasirekha, D. R. Vishnu, and K. Surya, "Recognition of handwritten text using long short term memory (LSTM) recurrent neural network (RNN)," *AIP Conf. Proc.*, vol. 2095, no. 1, p. 030011, Apr. 2019, doi: 10.1063/1.5097522.

[4] B. Annanurov and N. M. Noor, "A compact deep learning model for Khmer handwritten text recognition," *IAES Int. J. Artif. Intell. IJ-AI*, vol. 10, no. 3, Art. no. 3, Sep. 2021, doi: 10.11591/ijai.v10.i3.pp584-591.

[5] B. Annanurov and N. M. Noor, "Handwritten Khmer text recognition," in *2016 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*, Pune, India: IEEE, Dec. 2016, pp. 176–179. doi: 10.1109/wiecon-ece.2016.8009112.

[6] H. Meng and D. Morariu, "Khmer character recognition using artificial neural network," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*, Dec. 2014, pp. 1–8. doi: 10.1109/APSIPA.2014.7041824.

[7] S. S. Ullah, L. Gang, M. Riaz, A. Ashfaq, S. Khan, and S. Khan, "Handwritten Digit Recognition: An Ensemble-Based Approach for Superior Performance".

[8] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[9] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997, doi: 10.1109/78.650093.

[10] K. Cho *et al.*, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, A. Moschitti, B. Pang, and W. Daelemans, Eds., Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734. doi: 10.3115/v1/D14-1179.