I.T.C

Content list available at ITC

Techno-Science Research Journal

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

Techno-Science Research Journal

Empowering Education with Online Khmer Handwritten Text Recognition for Teaching and Learning Assistance

Heng Ham[1*], Dona Valy[2], Phutphalla Kong[2]

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

[2] *Department of Information and Communication Engineering, Institute of Technology of Cambodia, Russian Federation Blvd., P.O. Box 86, Phnom Penh, Cambodia*

Received: 17 July 2024; Revised: 25 July 2024; Accepted: 10 August 2024; Available online: 30 August 2025

Abstract: *The Khmer script presents a unique challenge due to its distinct characters, which make recognizing handwriting especially challenging. This study tackles this challenge by creating a specialized computer program that recognizes handwritten Khmer text. Our goal is to enhance Khmer language education by offering a valuable tool that supports educators and students in improving their Khmer language proficiency. This user-friendly tool improves handwriting skills and the program's adaptability and accuracy across handwriting styles, making learning more stimulating and effective. To accomplish this objective, we are beginning an extensive data collection initiative, creating a diverse dataset of handwritten Khmer samples. This dataset forms the foundation of our program's training, facilitating precise recognition. Despite the size of the dataset and the computational resource requirements, we are optimistic about this tool's potential impact on Khmer language education. Beyond improving handwriting, our program aims to improve Khmer educational material accessibility and efficiency. Utilizing a sequence-to-sequence-based model, particularly the LSTM-based encoder-decoder architecture, demonstrates our commitment to achieving the highest accuracy. Our model's performance is evaluated using the character error rate (CER), a metric that represents its precision in recognizing individual characters. With a dataset comprising 50,720 Khmer words, our model currently achieves an error rate of 3.36%, highlighting its effectiveness. By preserving and developing the Khmer language with advanced neural networks, our Khmer handwriting recognition tool preserves culture and promotes education in Cambodia. This tool has practical implications for enhancing teaching and learning experiences in educational settings by making educational materials more accessible and providing instant feedback on handwriting and text recognition.*

Keywords: Education; Khmer; Handwriting recognition; Long Short-Term Memory (LSTM)-based Encoder-Decoder

1. INTRODUCTION

Handwriting recognition is an incredible computer capability that converts handwritten text from scanned paper, photographs, or digital devices into editable text for computer-based text-processing applications. The evolution of online handwriting recognition is quite remarkable, with its origins in the 1960s, further development in the 1970s, and a resurgence in the 1980s [1,24]. This remains an ongoing challenge that continues to motivate advancements, driven by the need for improvement. Modern electronic tablets offer increased precision, while advanced algorithms contribute to systems with ever-enhancing accuracy. Recent approaches have moved towards recognizing entire sentences rather than the conventional method of segmenting and recognizing text in parts [4]. Current methodologies primarily employ machine learning due to its capability to recognize visual features, effectively surpassing the limitations of older feature engineering methods. A committed research community is actively involved in the field of handwriting recognition, with key conferences such as the International Conference on Frontiers in Handwriting Recognition (ICFHR) and the International Conference on

[*] Corresponding author: Heng Ham
E-mail: ham_heng@gsc.itc.edu.kh ; Tel: +855-81 359 150

Document Analysis and Recognition (ICDAR) serving as notable platforms for discussion. Online handwriting recognition has been thoroughly studied in a variety of scientific papers, with numerous detailed reviews published in previous research [19,23]. Also, recent work like the studies by Kim et al. [13] has brought new insights into the latest developments in this area, noting that "techniques are language and script-dependent." However, we believe that, although this is mostly true, many parts of these techniques can be adapted to work with multiple languages in a single recognition system. Principally, there are two methods used in researching online handwriting recognition: segment-and-decode (or over-segment and classify) [18], and analyzing the data in sequence [9].

Previous studies have explored into sequence-to-sequence learning with neural networks using varied methodologies and datasets. In a recent study by Sutskever et al. [22], the authors present a method for computers to comprehend and generate sequences of information, such as sentences in different languages. They utilize a specialized neural network known as Long Short-Term Memory (LSTM), which aids in retaining crucial details within a sequence. The authors establish a system where one component of the network comprehends the input sequence while another component generates the output sequence. This facilitates tasks like language translation. To educate the system, they utilize Maximum Likelihood Estimation (MLE)[17], which assists in teaching the computer how to produce accurate sequences. They demonstrate the effectiveness of their approach in machine translation, surpassing previous methods. Although attention mechanisms were not introduced in this paper, methods that focus on specific parts of a sequence laid the groundwork for subsequent improvements in sequence-to-sequence models. These methods have since become a fundamental component in many state-of-the-art natural language processing systems. Eisa et al. [6] propose using an LSTM encoder-decoder network with attention, significantly boosting accuracy well for reading handwritten words online. Their system effectively splits sentences into lines and demonstrates robust performance on the IAM On-Line Handwriting database. They suggest enhancing the system further with a spelling correction tool, and they achieved higher accuracy by using one-hot encoding. The system is versatile, performing well with various writers and operating efficiently without excessive computational requirements. It has the potential to be implemented in a wide range of applications. Nosouhian et al. [16] and their team developed a system to recognize handwritten words in real-time from pen movements. They used a recurrent neural network (RNN) with bidirectional long short-term memory (BiLSTM) and a smart guessing system, surpassing a hidden Markov model (HMM) in performance. Their system can interpret messy handwriting directly from pen movements, representing a major step forward in real-time handwriting recognition. The team's breakthrough technology has the potential to revolutionize the field of handwriting recognition. Liwicki et al. [15] developed a new method using a bidirectional recurrent neural network to read handwritten notes on whiteboards. This approach trains the computer to understand entire sequences of handwriting at once, leading to improved word recognition. They plan to enhance their system further by integrating a language model to recognize complete lines of text and improve performance with unfamiliar words, showing promise for advancing computer reading of handwritten notes. The goal is to reach a level of accuracy that rivals human capabilities. Gader et al. [7] developed a method to recognize handwritten numbers, focusing on ZIP codes from USPS mail. Their approach combines a template-based technique, which achieves better accuracy for most digits, with a model-based approach for the remaining digits. The system recognizes digits with high accuracy. This method proves reliable and effective for handwritten number recognition. This method proves reliable and effective for handwritten number recognition, with potential for further improvements.

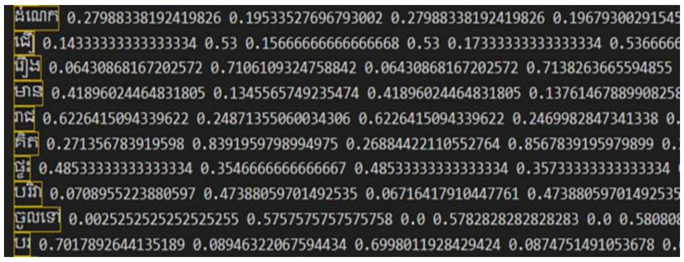
The Khmer script is the writing system used for creating texts in Khmer, the official language of Cambodia. Khmer is part of the Mon-Khmer group within the broader Austroasiatic language family. This script is well-known not only for its ancient history and beauty but also for the interesting challenges it poses to writers and font designers worldwide [10]. In modern Khmer script, there are 33 main consonant symbols, along with several less commonly used variants. Additionally, the script includes 17 dependent vowel symbols and about 14 special diacritic symbols that modify pronunciation and change the meaning of syllables. Moreover, certain consonants can be used as subscripts, resulting in a remarkable diversity of over 1,000 distinct ways to represent the Khmer language in written form today [2]. Khmer presents a unique challenge in handwriting due to its complex writing system. Currently, there is a lack of online tools capable of accurately recognizing and providing feedback on Khmer handwriting, limiting language learners. Developing a reliable Khmer handwriting recognition tool is difficult due to limited data availability and computational constraints for complex machine-learning algorithms. To address this need, there is an opportunity to create an effective and user-friendly online Khmer handwriting recognition tool that utilizes a substantial dataset to enhance accuracy. This tool could greatly benefit researchers, students, and professionals working with Khmer language documents.

The paper is structured into four primary sections. At first, it begins with an introduction that provides a broad overview of the research topic. Following the introduction, Section 2 details the specific methods employed in the study. Subsequently, Section 3 offers insight into the results obtained and their significance. Lastly, Section 4 concludes the paper by summarizing the main findings and discussing potential implications for future research.

2. METHODOLOGY

2.1 Dataset

In our study, a total of 50,720 Khmer words were collected from 25 traditional Khmer stories sourced from Wikipedia, as illustrated in Fig. 1. This figure represents the dataset we meticulously curated for our research. This collaborative endeavor engaged 28 individuals. Next, we employed the TKinterface framework along with Python to construct a customized dataset, as showcased in Fig. 2. This tool extracts text from .txt files and uses Khmer NLTK as a tokenizer to break the text into individual words, allowing us to analyze and visualize the word distribution. This dataset was segmented into three parts for training our handwriting recognition tool: 80% for model training to detect variations in handwritten Khmer characters, 10% for validation to improve the model and prevent overfitting, and the remaining 10% for testing to evaluate the model's performance on new data. This methodical approach optimizes both the learning process and the accuracy of the tool's evaluation. The dataset is structured as follows: [Label x1, y1, #,... xn, yn, #], where "label" represents the Khmer word and x1, y1,…, xn, yn denote the coordinates, with "#" indicating the end of a stroke.



```
ដំណេក 0.27988338192419826 0.19533527696793002 0.27988338192419826 0.1967930029154
ដើ 0.14333333333333334 0.53 0.15666666666666668 0.53 0.17333333333333334 0.536666
រឿង 0.06430868167202572 0.7106109324758842 0.06430868167202572 0.7138263665594855
មាន 0.41896024464831805 0.1345565749235474 0.41896024464831805 0.1376146788990825
ភាជ 0.6226415094339622 0.24871355060034306 0.6226415094339622 0.2469982847341338 0.
គិត 0.271356783919598 0.8391959798994975 0.26884422110552764 0.8567839195979899 0.
ផ្ទះ 0.48533333333333334 0.35466666666666667 0.48533333333333334 0.35733333333333334
បរិ 0.0708955223880597 0.47388059701492535 0.06716417910447761 0.47388059701492535
ចូលទៅ 0.0025252525252525255 0.5757575757575758 0.0 0.5782828282828283 0.0 0.58080
បរ 0.7017892644135189 0.08946322067594434 0.6998011928429424 0.0874751491053678 0.
```

Fig. 1. Khmer word dataset

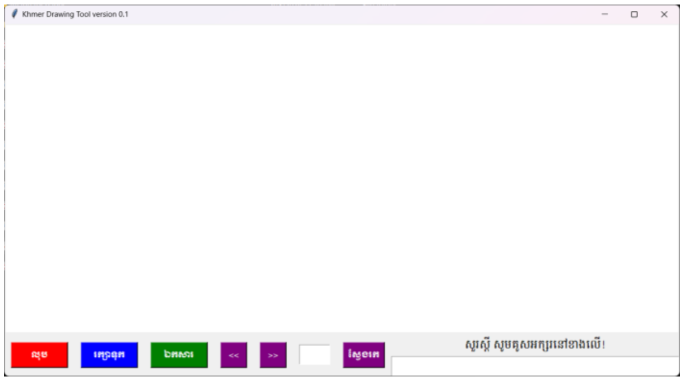


Fig. 2. Dataset creation tool

2.2 Data preprocessing

Before incorporating our dataset into our project, meticulous data quality assessments were conducted, which included verification of data consistency, accuracy, and completeness. Specifically, stringent measures were taken to verify that all Khmer words were exclusively in the Khmer language, thereby necessitating the removal of any unusual symbols or characters. Additionally, we visually inspected the dataset to confirm its clarity and fit for integration into our program. After ensuring the data was accurate, we analyzed it for mistakes or errors by comparing data points and checking for outliers. This thorough review process ensured that our program would receive reliable input for optimal performance. Finally, we ran several tests to confirm the integrity of the data and ensure it met our program's requirements. This careful approach to data preparation was crucial for the effectiveness of our analysis. By following these steps, we ensured that the data was clean and ready for processing. The detailed testing and validation process helped us achieve accurate results in our program.Additionally, we applied a normalization transformation to scale our data within the range of [0,1] and used a technique to segment strokes into sub-strokes.This involved calculating the mean of each stroke and dividing it into three parts. This segmentation allowed us to analyze the data more effectively and extract valuable information for our program. Overall, the combination of thorough data preparation, testing, and segmentation techniques greatly enhanced the accuracy and efficiency of our analysis.

2.3 Recurrent Neural Network

Recurrent neural networks (RNN) are a specific type of network that create memory using recurrent connections as shown in Fig. 3. Unlike feedforward networks, where inputs are unrelated to each other, RNNs establish connections among all inputs. This interconnectedness enables the network to demonstrate dynamic temporal patterns over a sequence of time, which is advantageous for tasks like sequential classification, such as sentiment analysis [3]. Initially, it takes x_0 from the input sequence and outputs h_0, which together with x_1, forms the input for the next step. Subsequently, h_0 and x_1 become the input for the subsequent step. Likewise, h_1 from the next step pairs with x_2 for the subsequent step, and this pattern continues. In this manner, the network retains context throughout the training process.

Based on mathematical formulations, we can describe the Recurrent Neural Network (RNN) using equations from Equation 1 to Equation 2. Equation 1 explains how the network updates its hidden state h_t over time by combining current input x_t previous state h_{t-1} , and biases with weighted transformations and an activation function f. Equation 2 shows how the RNN generates output y_t based on h_t using additional weights, a bias c and an activation function g.These equations outline how RNNs handle sequences of data to predict outcomes.

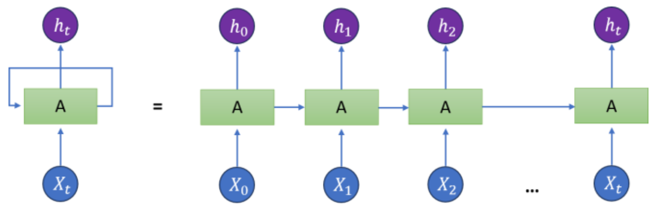


Fig. 3. Recurrent neural network loop

$$h_t = f(W_{hx}x_t + W_{hh}h_{t-1} + b) \quad \text{(Eq. 1)}$$

$$y_t = g(W_{yh}h_t + c) \quad \text{(Eq. 2)}$$

2.4 Long Short-Term Memory

In 1997, Sepp Hochreiter and Juergen Schmidhuber introduced an improvement to RNN called Long Short-Term Memory (LSTM) units [11], as illustrated in Fig. 4. General RNNs often encounter issues with error propagation over time, which LSTM units address by mitigating these error accumulations. By maintaining a more stable error flow, LSTMs enable RNNs to continue learning effectively over multiple time steps. LSTMs incorporate specialized gated blocks to manage and retain information crucial for learning, distinct from the typical flow of an RNN [25]. The specialized gated blocks in LSTMs include forget gates, input gates, and output gates, which serve different functions in controlling the flow of information. Neural network nodes are activated by the input they receive, similar to how LSTM gates selectively allow or block data based on their weighted inputs. These signals are then processed with specific sets of weights. As part of the learning process, RNNs adjust these weights that define hidden states and input behaviors. This adaptive mechanism allows the cells to learn when to accept, release, or discard information through iterative steps involving prediction, error backpropagation, and weight adjustment via gradient descent [8].

In the Long Short-Term Memory (LSTM) model, the formulation is shown in Equations 3 to 7. Equation 3 describes how the forget gate works. The forget gate decides which information from the previous cell state h_{t-1}should be carried over to the current cell state. The weights W_f *and* U_f, along with the bias b_f , are used to process the input x_t and the previous hidden state h_{t-1}.The function σ_g is the sigmoid activation function that outputs values between 0 and 1, indicating the extent to which each piece of information should be forgotten or retained. Equation 4 defines the input gate. The input gate decides which new information from the current input x_t and the previous hidden state h_{t-1} should be added to the cell state. The function $\sigma_g$is the sigmoid activation function indicating the importance of the new information. Equation 5 specifies the output gate operation. The output gate determines which parts of the cell state should be output as the hidden state. The function σ_g is the sigmoid activation function, which outputs values between 0 and 1, indicating the extent to which the information should be output. Equation 6 defines how the cell state c_t is updated over time. It combines the previous cell state c_{t-1} adjusted by the forget gate f_t , with the new candidate values i_t scaled by the input gate i_t and the updated values from the candidate cell state c_t scaled by the output gate o_t. Equation 7 specifies how the hidden state h_t is computed based on the cell state c_t and the output gate o_t. Here, o_t scales the cell state c_t , which is then passed through the activation function σ_h to produce the hidden state h_t.

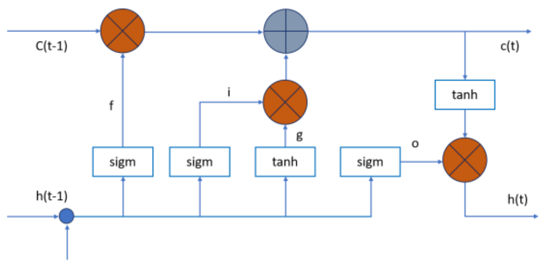


Fig. 4. Long-term short term memory unit

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \quad \text{(Eq. 3)}$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \quad \text{(Eq. 4)}$$

$$o_t = \sigma_g(W_0 x_t + U_o h_{t-1} + b_o) \quad \text{(Eq. 5)}$$

$$c_t = f_t o c_{t-1} + i_t o \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \quad \text{(Eq. 6)}$$

$$h_t = o_t o \sigma_h(c_t) \quad \text{(Eq. 7)}$$

2.5 Bidirectional LSTM

Bidirectional LSTMs are particularly useful when you need to understand the context of input, such as in sentiment analysis tasks [21]. Unlike regular LSTM models that only process information in one direction (from past to future), Bi-directional LSTMs process data in both directions, backward and forward, using two hidden states. This dual approach helps BiLSTMs understand context more effectively, allowing them to handle a larger part of input data. To visualize how BiLSTMs work, imagine splitting the neurons of a typical RNN into two paths: one for backward states (looking at earlier data) and one for forward states (looking at future data). These pathways do not share inputs directly, and a diagram of this BiLSTM structure is shown in Fig. 5. By adopting bidirectional processing, Bidirectional LSTMs can incorporate input data from both past and future time frames simultaneously. This contrasts with standard RNNs, which typically require delays to include future data in their computations. [21]

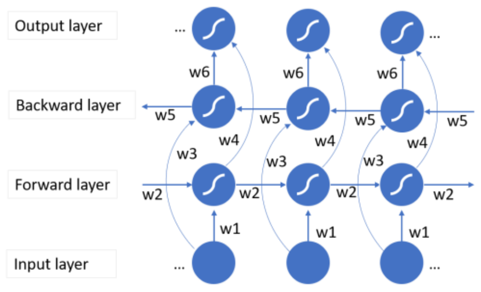


Fig. 5. Bidirectional LSTM

2.6 Encoder–decoder networks

The encoder-decoder framework was introduced to address machine translation challenges involving sentences in different languages. This approach, developed in 2014 by Ilya Sutskever and colleagues [22], uses a multilayered LSTM to convert the input sequence into a fixed-dimensional vector. Then, another deep LSTM decodes this vector to produce the target sequence. Fig. 6 illustrates this process. This method has been demonstrated to be highly successful and is now widely applied in various text-related tasks, such as summarization. It is also considered state-of-the-art in online handwriting recognition and forms the core architecture of the Google translation service [12].

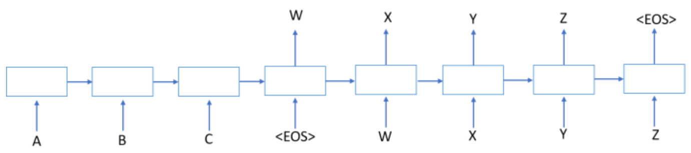


Fig. 6. Encoder-decoder for sequence translation

2.7 Encoder-Decoder LSTM architecture

The model we propose is an Long Short-Term Memory encoder-decoder architecture inspired by the work in reference [5], as illustrated in Fig. 7. The encoder component is composed of a BiLSTM layer, which processes the input sequence by capturing both forward and backward dependencies. This mechanism allows the encoder to create a context vector by combining the hidden states from both the forward and backward LSTMs. Subsequently, this context vector, along with the encoder's hidden state, is forwarded to the decoder. The decoder, in turn, employs another LSTM layer to generate the output sequence, applying the contextual information obtained from the encoder. Finally, the model produces the desired output sequence by passing the decoder's output through an output layer. This approach utilizes the strengths of LSTMs to efficiently handle sequential data and incorporates external static features to enhance the overall predictive capabilities. By combining the bidirectional nature of LSTMs with the encoder-decoder architecture, the model can effectively capture long-range dependencies in the input sequence. This comprehensive approach allows for more accurate predictions by leveraging both temporal information and external features.

2.8 Evaluation Metrics

For our evaluation, we will utilize the Character Error Rate (CER) as our metric, which is suitable for predicting Khmer words. The CER calculates the percentage of incorrect characters by dividing the minimum edit distance between the predicted word and the ground truth by the total number of characters in the ground truth, and then multiplying the result by 100% (Eq. 8):

$$CER= (eD / NC) \times 100\% \qquad \text{(Eq. 8)}$$

where:
eD = minimum edit distance
NC = number of characters

The minimal edit distance, commonly known as the Levenshtein distance, is calculated by estimating the fewest number of editing operations necessary to convert one string to another. These operations are insertion (adding a character), deletion (removing a character), and substitution (replacing one character with another)[20]. To ensure efficiency, this distance is often calculated by dynamic programming. By calculating the CER, we can efficiently compare the accuracy of our projected Khmer words to the ground truth.

3. RESULTS AND DISCUSSION

By employing the Grid Search technique [14] to fine-tune the hyperparameters of our model, we were able to identify the optimal values, as demonstrated in Table 1. This meticulous process resulted in a significant reduction of the error rate to 3.36% on the test dataset, as illustrated in Table 2. The model architecture consisted of three layers, and we trained it for 100 epochs while configuring key parameters: a hidden size of 1024, a batch size of 32, a dropout rate of 0.2, and a learning rate of 0.001, all implemented within the PyTorch framework. Additionally, we utilized the Adam optimizer and implemented early stopping techniques to prevent overfitting during training. These adjustments significantly improved performance compared to our initial results. Furthermore, our study involved processing a dataset of 50,720 Khmer words, which presented unique challenges due to the complexity of Khmer script and handwriting styles. Despite starting with an error rate of 3.36%, the refined model demonstrated significant advancements in Khmer handwriting recognition technology. This progress signifies a promising development in improving the accuracy and efficiency of handwritten text recognition and enhancing the accessibility and usability of Khmer language resources for learners and users.

Table 1. Randomized Hyperparameter Tuning

Hyperparameter	Number of Elements
Learning rate	[0.1, 0.01, 0.001, 0.0001]
Hidden size	[16,32,64,128,256,512,1024]
Batch Size	[16,32,64,128,256,512]
Dropout	[0.1, 0.2, 0.3, 0.4, 0.5]
Number of Layer	[1,2,3,4,5,6,7,8,9,10]
Number of epoch	[50,100]

The success of our refined model highlights the potential for further advancements in Khmer handwriting recognition technology, paving the way for more accurate and dependable technologies in the future. By continuing to refine and enhance the model, we can strive towards even greater accuracy and usability in Khmer handwriting recognition technology. This will not only benefit learners and users of the Khmer language but also contribute to the overall advancement of handwriting recognition technology as a whole. Furthermore, increased accuracy in Khmer handwriting recognition technology can lead to more efficient communication and information retrieval for individuals who rely on this language. Ultimately, these advancements can help bridge the gap between different linguistic communities and promote inclusivity in digital spaces.

Table 2. Results on the Train, Test, and Validation Sets

Hyperparameter	Value	CER on Train	CER on Test	CER on Valid
Learning rate	0.001			
Hidden size	1024			
Batch Size	32	1.14%	3.36%	3.22%
Dropout	0.2			
Number of Layer	3			
Number of epoch	100			

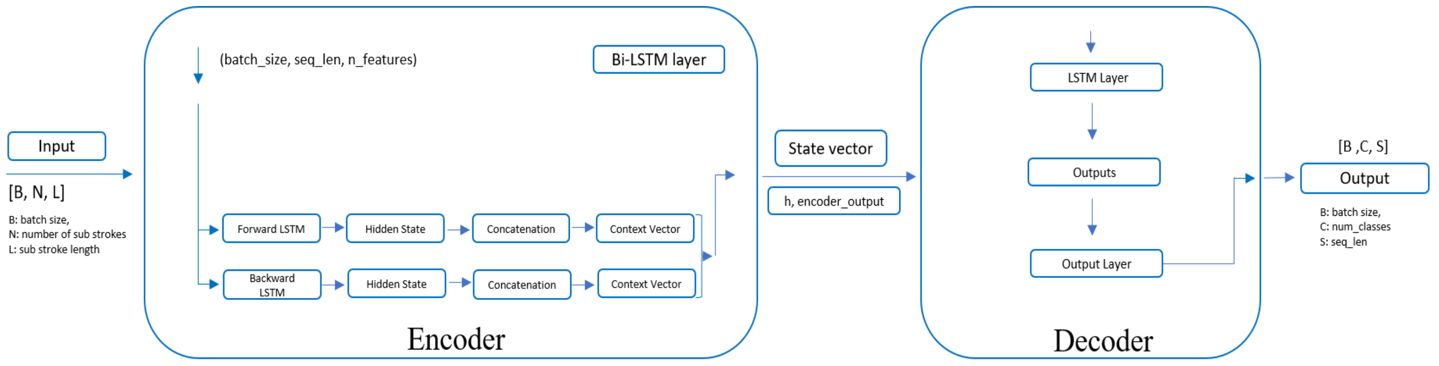


Fig. 7. LSTM-based encoder-decoder architecture

4. CONCLUSIONS

In conclusion, our research has resulted in the development of an advanced computer application specifically designed to identify handwritten Khmer text, aimed at enhancing Khmer language education. Despite the challenges we faced, we stayed optimistic about its positive impact on education. By improving the accessibility and efficiency of Khmer educational resources, our tool plays a critical role in preserving the Khmer script and cultural heritage. Utilizing state-of-the-art technologies like neural networks, we are pushing the boundaries of language recognition and supporting initiatives to recover the Khmer language. We think that the Khmer language community will gain a lot from our research in the years to come. Looking ahead, our future plans involve expanding our dataset and conducting further experiments with different models to further improve our program. Our objective is to continuously enhance its effectiveness and accuracy in recognizing handwritten Khmer text. Through ongoing research and development efforts, we are committed to improving our tool to better serve the Khmer language community and contribute to its revitalization efforts. By gathering more data and exploring innovative approaches, we are committed to ensuring that our application develops into a highly effective resource for promoting Khmer language education and preservation.

ACKNOWLEDGMENTS

We are deeply grateful to the Computer Vision and Natural Language Processing Research Laboratory at the Institute of Technology of Cambodia for their generous support and for hosting this research project.

REFERENCES

[1] Almuallim, H., & Yamaguchi, S. (1987, September). A Method of Recognition of Arabic Cursive Handwriting. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-9(5), 715–722. https://doi.org/10.1109/tpami.1987.4767970

[2] Annanurov, B., & Noor, N. M. (2016, December). Handwritten Khmer text recognition. 2016 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE). https://doi.org/10.1109/wiecon-ece.2016.800911

[3] Aziz Sharfuddin, A., Nafis Tihami, M., & Saiful Islam, M. (2018, September). A Deep Recurrent Neural Network with BiLSTM model for Sentiment Classification. 2018 International Conference on Bangla Speech and Language Processing(ICBSLP).https://doi.org/10.1109/icbslp.2018.8554396

[4] BIADSY, F., SAABNI, R., & EL-SANA, J. (2011, November). SEGMENTATION-FREE ONLINE ARABIC HANDWRITING RECOGNITION. International Journal of Pattern Recognition and Artificial Intelligence, 25(07), 1009–1033. https://doi.org/10.1142/s0218001411008956

[5] Born, S., Valy, D., & Kong, P. (2022, December 2). Encoder-Decoder Language Model for Khmer Handwritten Text Recognition in Historical Documents. 2022 14th International Conference on Software, Knowledge, Information Management and Applications (SKIMA). https://doi.org/10.1109/skima57145.2022.10029532

[6] Eisa, A., Abdalla, L., & Ahmed, M. (2019). Online Handwriting Recognition Using Encoder-Decoder Model. 2019 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE),1–7. https://doi.org/10.1109/ICCCEEE46830.2019.9071037

[7] Gader, P., Forester, B., Ganzberger, M., Gillies, A., Mitchell, B., Whalen, M., & Yocum, T. (1991). Recognition of handwritten digits using template and model matching. Pattern Recognition, 24(5), 421–431. https://doi.org/10.1016/0031-3203(91)90055-A

[8] Gers, F. A., Schmidhuber, J., & Cummins, F. (2000, October 1). Learning to Forget: Continual Prediction with LSTM. Neural Computation, 12(10), 2451–2471. https://doi.org/10.1162/089976600300015015

[9] Graves, A., Liwicki, M., Fernandez, S., Bertolami, R., Bunke, H., & Schmidhuber, J. (2009, May). A Novel Connectionist System for Unconstrained Handwriting Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(5), 855–868. https://doi.org/10.1109/tpami.2008.137

[10] Haralambous, Y. (1994). Typesetting Khmer. Electronic Publishing, 7(4), 197-216.

[11] Hochreiter, S., & Schmidhuber, J. (1997, November 1). Long Short-Term Memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[12] Keysers, D., Deselaers, T., Rowley, H. A., Wang, L. L., & Carbune, V. (2017, June 1). Multi-Language Online Handwriting Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6), 1180–1194. https://doi.org/10.1109/tpami.2016.2572693

[13] Kim, J., & Sin, B. K. (2014). Online Handwriting Recognition. Handbook of Document Image Processing and Recognition, 887–915. https://doi.org/10.1007/978-0-85729-859-1_29

[14] Liashchynskyi, P., & Liashchynskyi, P. (2019). Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS. arXiv (Cornell University). https://doi.org/10.48550/arxiv.1912.06059

[15] Liwicki, M., Graves, A., Schmidhuber, J., & Bunke, H. (n.d.). A Novel Approach to On-Line Handwriting Recognition Based on Bidirectional Long Short-Term Memory Networks.

[16] Nosouhian, S., Nosouhian, F., & Kazemi Khoshouei, A. (2021). A Review of Recurrent Neural Network Architecture for Sequence Learning: Comparison between LSTM and GRU [Preprint]. other. https://doi.org/10.20944/preprints202107.0252.v1

[17] Pan, JX., Fang, KT. (2002). Maximum Likelihood Estimation. In: Growth Curve Models and Statistical Diagnostics. Springer Series in Statistics. Springer, New York, NY. https://doi.org/10.1007/978-0-387-21812-0_3

[18] Pittman, J. A. (2007, September). Handwriting Recognition: Tablet PC Text Input. Computer, 40(9), 49–54. https://doi.org/10.1109/mc.2007.314

[19] Plamondon, R., & Srihari, S. (2000). Online and off-line handwriting recognition: a comprehensive survey. IEEE Transactions on Pattern Analysis and Machine Intelligence,22(1),6384. https://doi.org/10.1109/34.824821

[20] Rani, S., Singh, J. (2018). Enhancing Levenshtein's Edit Distance Algorithm for Evaluating Document Similarity. In: Sharma, R., Mantri, A., Dua, S. (eds) Computing, Analytics and Networks. ICAN 2017. Communications in Computer and Information Science, vol 805. Springer, Singapore. https://doi.org/10.1007/978-981-13-0755-3_6

[21] Schuster, M., & Paliwal, K. (1997). Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, 45(11), 2673–2681. https://doi.org/10.1109/78.650093

[22] Sutskever, I., Vinyals, O., & Le, Q. (2014, September 10). Sequence to Sequence Learning with Neural Networks.arXiv.org.https://doi.org/10.48550/arXiv.1409.3215

[23] Tappert, C., Suen, C., & Wakahara, T. (1990). The state of the art in online handwriting recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence,12(8),787808.https://doi.org/10.1109/34.57669

[24]] Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., & Lang, K. (1988). Phoneme recognition: neural networks vs. hidden Markov models vs. hidden Markov models. ICASSP-88., International Conference on Acoustics, Speech, and Signal Processing. https://doi.org/10.1109/icassp.1988.196523

[25] Wang, J., Yu, L. C., Lai, K. R., & Zhang, X. (2016). Dimensional Sentiment Analysis Using a Regional CNN-LSTM Model. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). https://doi.org/10.18653/v1/p16-2037