State-of-the-Art Khmer Text Recognition Using Deep Learning Models

Saly Keo[†‡] Mickaël Coustaty[†] Souhail Bakkali[†] Marçal Rossinyol[†]
[†]La Rochelle University, Laboratoire Informatique Image Interaction (L3i)
[‡]Cambodia Academy of Digital Technology (CADT)
{keo.saly,mickael.coustaty,souhail.bakkali}@univ-lr.fr,
marcal@allread.ai

Abstract

This paper explores the state-of-the-art methods for Khmer Optical Character Recognition (OCR), highlighting the challenges and opportunities in recognizing this complex script. While OCR systems have shown great success in languages like English, the Khmer script presents unique difficulties due to its complex structures, lack of word boundaries, and variety of shapes and sizes in characters. In this work, we evaluate three prominent text recognition models: Trans + ResNet + BiLSTM + Attention (TRBA), Convolutional Recurrent Neural Network (CRNN), and Tesseract OCR. A comparative analysis was conducted using a dataset consisting of 284,098 word images, split into training, validation, and testing sets. We assess the performance of each model using Character Error Rate (CER) and Word Error Rate (WER). The results show that the TRBA model achieves the highest accuracy, outperforming CRNN and Tesseract OCR. We also present future directions to further enhance Khmer OCR systems using deep learning techniques.

Keywords: Khmer OCR, Optical Character Recognition, Khmer Deep Learning

1 Introduction

Optical Character Recognition (OCR) is a technology that converts images of handwritten or printed text into a digital format that computers can understand and process. This process allows scanned documents, photos of text, or any text-based images to be converted into editable and searchable text files, making it easier to store, retrieve, and work with the information [1]. OCR systems are widely used across various applications. For example, when given an image of a book or newspaper, an OCR system can automatically read and convert the text into a sequence of ASCII or Unicode characters. These characters can then be utilized for different tasks, such as searching, highlighting, annotating, or translating the text. Furthermore, OCR technology is used in many different areas, both in government and private organizations. For example, the immigration department uses OCR to quickly and accurately recognize passports, making the identification process faster and more secure. City Traffic Police also use OCR to automatically recognize vehicle number plates, helping to manage traffic and monitor vehicles more effectively. In the private sector, OCR is valuable in banking for processing checks, in healthcare for digitizing patient records, and in retail for automating invoice processing. This makes OCR a versatile tool that simplifies tasks and improves accuracy across many fields [2].

In recent years, advances in artificial intelligence (AI) have accelerated research in the OCR field, leading to the development of significantly more accurate OCR technology. While OCR has been highly effective for English and other high-resource languages, with development spanning over eight decades, the early work on OCR for the Khmer language only began around 2005 [3]. Khmer is the official language of Cambodia and is spoken by around 16 million people. It uses its own alphabet and writing system. In Khmer, words are made up of syllables that combine consonants and vowels. The Khmer alphabet has 33 consonants. This writing system is special and reflects the unique language of Cambodia. Khmer scripts are more complex than Latin-based writing systems. This is because Khmer has unique features in its layout. Despite its significance, Khmer OCR systems face unique challenges due to the complex script and linguistic characteristics of the language, as shown in Figure 1. The Khmer writing system is intricate, with consonant clusters being one of the primary features that complicate OCR tasks. One or two consonants can be stacked below an initial con-

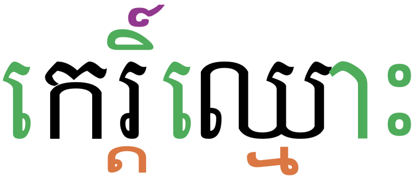


Figure 1. An example of a complex Khmer word that combines all parts, including consonant (black), vowel (green), subscript (orange), and diacritics (purple)

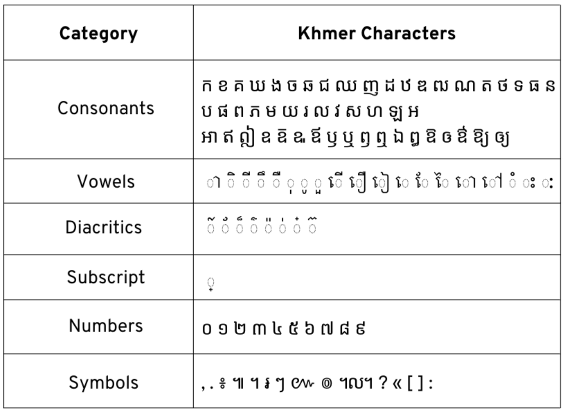
sonant in a form known as ”Coeng” (meaning foot in English), creating a consonant cluster [3]. These stacked consonants modify the base consonant, creating new sounds, which OCR systems must correctly identify and interpret. In addition to consonant stacking, Khmer writing makes use of diacritical marks (diacritics) that are placed above consonants. These diacritics alter the sound of the consonant or vowel and are a critical aspect of correct text interpretation. Moreover, Khmer employs dependent vowels, which cannot stand alone. These vowels must always be attached to a consonant and are placed in various positions relative to the consonant—on the left, right, above, below, or even surrounding the base consonant. This positional flexibility further increases the complexity for OCR systems, as they need to accurately capture the entire combination of consonant, diacritic, and vowel placement.

Additionally, Khmer script is written without spaces between words, making it difficult for OCR systems to segment text accurately. The lack of clear word boundaries adds to the challenge of identifying where words begin and end, requiring the system to rely on language rules and character patterns. Because of these unique features, Khmer script requires a much more complex rendering layout compared to Latin-based writing systems. Our study not only explores these existing solutions but also involves pre-training on available datasets to improve the accuracy and performance of Khmer OCR systems.

2 Related Works

2.1 Khmer Text Recognition

In the early development of Khmer OCR, [4] proposed a method to extract information from images and create templates for characters using wavelet analysis. This technique involves extracting data from an image and comparing it with all training templates. According to the authors, their method achieved accuracy rates of 92.85%, 91.66%, and 89.27% for 22-point, 18-point, and 12-point fonts, respectively.



Category	Khmer Characters
Consonants	ក ខ គ ឃ ង ច ឆ ជ ឈ ញ ដ ឋ ឌ ឍ ណ ត ថ ទ ធ ន ប ផ ព ភ ម យ រ ល វ ស ហ ឡ អ អា ឥ ឦ ឧ ឱ ឩ ឪ ឫ ឬ ឭ ឮ ឯ ឰ ឱ ឲ ឳ ឱ្យ ឲ្យ
Vowels	◌ា ◌ិ ◌ី ◌ឹ ◌ឺ ◌ុ ◌ូ ◌ួ ◌ើ ◌ឿ ◌ៀ ◌េ ◌ែ ◌ៃ ◌ោ ◌ៅ ◌ំ ◌ះ ◌ៈ
Diacritics	◌៉ ◌៊ ◌៍ ◌៌ ◌៎ ◌់ ◌៏ ◌័
Subscript	◌្
Numbers	០ ១ ២ ៣ ៤ ៥ ៦ ៧ ៨ ៩
Symbols	, . ៖ ៕ ។ ៛ ៗ ៚ ៙ ។ល។ ? « [ ] :

Figure 2. Khmer Characters

Regarding to [5] The research implemented an SVM-based classification system for Khmer characters, utilizing three different SVM kernels—Gaussian, Polynomial, and Linear—on training and recognition tasks to identify the most effective kernel for the Khmer language. This approach enabled the use of a small training dataset by dividing character training into smaller segments rather than processing large clusters. The classification process employed binary data, with 0 representing white space and 1 representing the character’s black pixel area. Each segment was reshaped into a matrix of binary data across various image sizes, from which features were extracted for SVM classification. Post-recognition, certain rules were applied to combine clusters or characters, including character-level analysis and common mistake corrections. Experiments on approximately 750 pure Khmer words (around 3,000 characters) demonstrated that the SVM method with the Gaussian Kernel delivered superior performance compared to the other kernels. The system trained on a single font, ”Khmer OS Content” at 32pt, and achieved recognition accuracy rates about 98% for various font sizes. The system’s effectiveness depended on the character segmentation process, which utilized edge detection. As a result, the proposed OCR system was not well-suited for handling noisy text-line images.

The Khmer Character Recognition system incorporates two artificial neural network techniques: a self-organizing map and a multilayer perceptron with a backpropagation algorithm, as described by [6]. The system was trained on a predefined dataset consisting of five font sets, each containing 33 characters and 10 numerals. The training process involved two phases: first, using a self-organizing map network, followed by a second phase with a multilayer perceptron network. This approach reduced complexity and accelerated character recognition, as preclassification directed specific modules of the multilayer perceptron classifiers. The system achieved an average recognition accuracy of 65% on the trained dataset, while the accuracy on untrained datasets dropped to approximately 30% due to noise.

The training system utilizing the HTK Toolkit is demonstrated by [7], the authors focused specifically on the Limon S1 Khmer font at size 22 because it is commonly used in Khmer documents. Their OCR system involves four steps. The first step is pre-processing, which includes line separation. In the second step, character blocks are segmented into atomic shapes such as Main Body, SuperScript, SubScript, CCDown, and Complex Characters (CC). The third step involves recognition, where segmented shapes are sent to a recognizer that assigns an ID to each shape. The final step is mapping, which converts the ID into an ASCII code using a code file that lists unique IDs and their corresponding ASCII codes. The system achieved an average recognition rate of 96.34% for all shapes. Its performance depended on character separation, which used vertical white space as a delimiter.

Convolutional Neural Networks (CNNs) were introduced by [8], with the aim to develop a recognition system for Khmer handwritten. This study includes six sets of handwriting samples, each containing 33 consonants (root radicals) and 17 vowels, totaling 561 syllables. The use of CNNs led to a recognition rate of 94.85% for Khmer handwriting. However, the experiments conducted so far are limited to Khmer consonants, with further work needed on vowels, numerals, and other symbols.

An end-to-end deep convolutional recurrent neural network solution for Khmer optical character recognition has been proposed by [3], this research employs a sequence-to-sequence (Seq2Seq) architecture with an attention mechanism. The encoder extracts visual features from input text-line images using layers of convolutional blocks and a gated recurrent unit (GRU) layer. The Seq2Seq Khmer OCR network is trained on a large dataset of computer-generated text-line images featuring multiple common Khmer fonts, with complex data augmentation applied to both the training and validation datasets. The proposed model significantly outperforms the state-of-the-art Tesseract OCR engine for the Khmer language, achieving a character error rate (CER) of 0.7% compared to Tesseract's 35.9% on a validation set of 6,400 augmented images.

Regarding to [9], conducted a baseline experiment on isolated character recognition using a multilayer convolutional neural network. They introduced the SleukRith Set, the first dataset comprised of digital images of Khmer palm leaf manuscripts, gathered both from their own digitization efforts and existing digital content from various sources. The dataset includes 657 manuscript pages, and a tool was developed to annotate these pages, resulting in three types of data: an isolated character dataset, an annotated word dataset, and line segmentation ground truth. The authors reported that the network achieved an error rate of 6.04%, indicating that there is still potential for improvement.

The Convolutional Transformer-based text recognition method tailored for low-resource non-Latin scripts, utilizing local two-dimensional (2D) feature maps was introduced by [10]. This method can process images of arbitrarily long text lines, common in non-Latin writing systems without explicit word boundaries, without the need to resize them to a fixed size, thanks to an enhanced image chunking and merging strategy. The research was conducted on synthetically generated datasets, including 1.5 million textline images with plain white backgrounds for document OCR training and 1.3 million randomly augmented images for synthetic scene text training. The study highlights that across all evaluated scripts, the proposed 2D models consistently outperformed baseline models like Tesseract OCR, CRNN, TRBC, and TRBA, achieving lower character error rates (CERs) compared to the 1D models.

3 Proposed Method

In this section, we review and suggest a deep learning architecture that has shown strong performance for Khmer Optical Character Recognition (OCR). This model integrates **ResNet** for feature extraction, **BiLSTM** for sequence modeling, and an **Attention Mechanism** for refining predictions. It addresses the unique challenges of Khmer OCR, such as the lack of word boundaries, complex syllabic structures, and variability in character shapes and sizes. However, We did not apply **Transformation** to normalize the input text images in this work, as our dataset consists of printed text that is already clean.

3.1 Key Components of the Model

3.1.1. Feature Extraction Stage: The model uses **ResNet**, a Convolutional Neural Network (CNN) architecture known for its residual connections, to extract key visual features from the input images. ResNet is highly effective in learning complex patterns such as edges and curves, which is crucial for distinguishing between visually similar Khmer characters.

3.1.2. Sequence Modeling Stage: The model incorporates **Bidirectional Long Short-Term Memory (BiLSTM)** networks to capture sequential dependencies in text. This is essential for Khmer OCR, where the meaning of a character often depends on its position within a word. BiLSTM allows the model to understand the full context of each character in a sequence.

3.1.3. Prediction Stage with Attention Mechanism: To further refine predictions, an **Attention Mechanism** is used to focus on the most relevant parts of the text image during recognition.

3.2 The TRBA Model

The reviewed architecture, TRBA demonstrates strong performance by combining:

- **Trans** for standardized, or ”normalized,” to ensure consistency in shape and size, as the input images often vary. To handle this, we applied the thin-plate spline (TPS) transformation, a method that adjusts the image to a uniform shape.

- **ResNet** for robust feature extraction from Khmer characters.

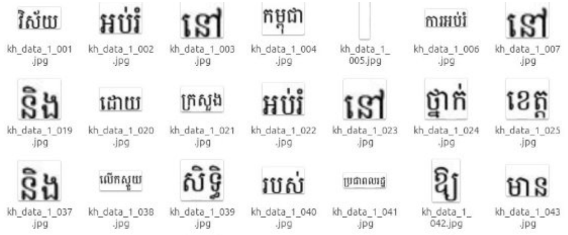


Figure 3. The Kaggle dataset sample is cropped at the word level.

- **BiLSTM** to model the sequence of characters, capturing dependencies in both directions.

- **Attention Mechanism** to focus on critical regions of the image, especially in visually complex areas.

3.3 Training Strategy

The model was trained on the **Khmer Dataset for Word Spotting**, consisting of over 284,098 word images, split into training, validation, and testing sets (80%, 10%, 10%).

3.4 Model Comparison

In our evaluation, the TRBA model outperformed two other models:

- **Convolutional Recurrent Neural Network (CRNN)**: A commonly used baseline model for OCR tasks.

- **Tesseract OCR**: An open-source OCR engine.

The TRBA model achieved the best results in terms of **Character Error Rate (CER)** and **Word Error Rate (WER)**, demonstrating its effectiveness for Khmer OCR compared to CRNN and Tesseract OCR.

4 Experiments and Analysis

For training and evaluating the model, we employed the Khmer Dataset for Word Spotting, which is available on Kaggle [1]. This dataset consists of Khmer printed documents with sentence-based images, and includes word-level labels in a corresponding XML file. It contains a total of 3,376 sentence images, which equates to

[1] https://www.kaggle.com/datasets/keatchakravuth/khmer-annotation/data

284,098 individual word images as shown in figure 3. The dataset was divided into three categories: training, validation, and testing sets, as detailed in Table 1.

Table 1. Sample Dataset.

Word Level	Number of Image
Training	227,838
Validation	28,130
Testing	28,130

To evaluate how well deep learning techniques can identify text from images, we use two key metrics: Character Error Rate (CER) and Word Error Rate (WER). A lower score in these metrics means the technique is more effective, while a higher score indicates less accuracy. We use CER and WER because they are widely accepted and commonly used in OCR and speech recognition. These measures are popular because they provide a clear and straightforward way to evaluate how well a system recognizes text. By using CER and WER, we can easily compare the performance of different systems and see how accurate they are [11] [12].These metrics are built on the Levenshtein distance [13], which calculates how similar two strings are. They are defined as follows:

CER is calculated by dividing the Levenshtein distance between the actual and predicted characters by the maximum length of the actual and predicted character sequences. For an image I, let G(C, I) represent the ground truth characters and P(C, I) represent the predicted characters. Lev(G, P) is the Levenshtein distance between G(C, I) and P(C, I) [12]. The formula for CER_I is given by:

$$CER_I = \frac{LEV_{(C,I)}}{\max(|G_{(C,I)}|, |P_{(C,I)}|)}$$

A testing set S comprising N text-line or word images is used, and the CER_M is defined as follows:

$$CER_M = \frac{1}{N} \sum_{I=0}^{n} CER_I$$

WER is the ratio between the Levenshtein distance of the actual and predicted words and the length of the longer word excerpt. For an input image, G(W, I) represents the actual text, P(W, I) represents the predicted text, and Lev(G, P) is the Levenshtein distance between them. The WER for the image is calculated as follows:

$$WER_I = \frac{LEV(W,I)}{\max(|G(W,I)|, |P(W,I)|)}$$

A set S is used for testing, containing N text-line or word images. The WER_M is defined as follows:

$$WER_M = \frac{1}{N} \sum_{I=0}^{n} WER_I$$

4.1 Experimental Setup

For our experiment, we used the deep text recognition benchmark introduced by [14]. This benchmark breaks down the text recognition process into four key stages, which are common across many Scene Text Recognition (STR) models. These stages are as follows:

1. Transformation stage (Trans.): In this stage, the images are standardized, or ”normalized,” to ensure consistency in shape and size, as the input images often vary. To handle this, we applied the thin-plate spline (TPS) transformation, a method that adjusts the image to a uniform shape. TPS is a variant of the spatial transformer network (STN) [15], known for its flexibility in managing text lines with different aspect ratios. By using this transformation, we reduce the complexity caused by different image shapes, making it easier for the model to process.

2. Feature extraction stage (Feat.): This is the stage where the model extracts key features from the input image, focusing on what is important for recognizing the text. During this process, irrelevant details like font style, color, size, and background are ignored. To achieve this, we used ResNet, a convolutional neural network (CNN) that includes ”residual connections.” These connections help with training deeper neural networks, making it easier to extract detailed and meaningful features from the image.

3. Sequence modeling stage (Seq.): In this stage, the model captures the relationships between characters in a sequence. This means it looks at the context, so each character is recognized based not just on its appearance, but also based on the characters around it. This helps make the recognition of each character more accurate, especially when the characters are part of a sequence, like in words or sentences. Instead of recognizing characters in isolation, this step allows the model to consider the overall structure of the text.

4. Prediction stage (Pred.): The final stage is where the model predicts the sequence of characters. Based on the features extracted earlier, the model generates the final output, which is the recognized text. This stage turns the abstract features of the image into a readable string of characters that represents the text within the image.

Experiments are performed using two state-of-the-art deep learning techniques and gooogle tesseract OCR.

For the experiment, three models were used: TRBA, CRNN, and Tesseract OCR. The dataset consisted of 284,098 word images, which were randomly split into three parts. Specifically, 80% of the images were used for training the models, 10% were set aside for validation, and the remaining 10% were used for testing.

After performing a comparative analysis of the results, it was found that the TRBA model achieved the best performance, with the lowest Character Error Rate (CER) and Word Error Rate (WER). The TRBA model recorded a CER of 0.0007 and a WER of 0.0029, indicating highly accurate text recognition.

Next, the CRNN model also showed strong results, though slightly less accurate than TRBA. It achieved a CER of 0.0010 and a WER of 0.0044, which is still competitive.

Finally, Tesseract OCR, an open-source optical character recognition engine, performed less effectively in comparison to the deep learning models. It recorded a CER of 0.2082 and a WER of 0.4155, which were significantly higher than both the TRBA and CRNN models.

In summary, the TRBA model had the best performance, followed closely by CRNN, while Tesseract OCR showed the lowest accuracy in both CER and WER metrics.

Table 2. The comparative model for word level results

Technique	Average CER	Average WER
Tesseract OCR	0.2082	0.4155
CRNN	0.0010	0.0044
TRBA	0.0007	0.0029

5 Discussions and Future Works

The results from our experimentation highlight several key findings. TRBA model demonstrated superior performance in both Character Error Rate (CER) and Word Error Rate (WER) metrics, significantly outperforming both the CRNN model and Tesseract OCR. This indicates that deep learning models, particularly those using attention mechanisms, are highly effective at recognizing Khmer script, which has complex and unique character structures. The CRNN model also performed well, though not as accurately as TRBA, while Tesseract OCR struggled to handle the intricacies of the Khmer script, resulting in much higher error rates.

Despite these successes, there are still challenges that remain. First, while the TRBA model achieved low CER and WER scores, its performance may still be improved by training on larger, more diverse datasets that include handwritten and historical documents, in addition to printed text. Furthermore, exploring more sophisticated data augmentation techniques could help the models generalize better to unseen data, such as noisy images or images with various fonts and styles.

In conclusion, while the results from our experiments are promising, there is still room for improvement in Khmer OCR systems. By incorporating advanced neural architectures, expanding training datasets, and applying modern deep learning techniques such as transformers and transfer learning, we believe the accuracy of Khmer OCR can be further enhanced, paving the way for its broader application in government, education, and other industries.

Acknowledgment

I would like to express my gratitude to the University of La Rochelle for providing the server and support necessary to train the model and complete this research. I also extend my thanks to the Cambodia Academy of Digital Technology (CADT) for funding and supporting my studies. Additionally, I am grateful for the access to the Khmer Dataset for Word Spotting, which was essential for our experiments. Special thanks to the research community for their valuable contributions to the advancement of Khmer OCR technology.

References

[1] Robert H Davis and J Lyall. Recognition of handwritten characters—a review. *Image and vision computing*, 4(4):208–218, 1986.

[2] Gabriel Resende Gonçalves, Matheus Alves Diniz, Rayson Laroca, David Menotti, and William Robson Schwartz. Real-time automatic license plate recognition through deep multi-task networks. In *2018 31st SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)*, pages 110–117. IEEE, 2018.

[3] Rina Buoy, Nguonly Taing, Sovisal Chenda, and Sokchea Kor. Khmer printed character recognition using attention-based seq2seq network. *Ho Chi Minh City Open University Journal Of Science-Engineering And Technology*, 12(1):3–16, 2022.

[4] C Chey, Pinit Kumhom, and Kosin Chamnongthai. Khmer printed character recognition by using wavelet descriptors. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 14(03):337–350, 2006.

[5] Pongsametrey Sok and Nguonly Taing. Support vector machine (svm) based classifier for khmer printed character-set recognition. In *Signal and information processing association annual summit and conference (APSIPA), 2014 Asia-Pacific*, pages 1–9. IEEE, 2014.

[6] Hann Meng and Daniel Morariu. Khmer character recognition using artificial neural network. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*, pages 1–8. IEEE, 2014.

[7] Ahmed Muaz LengIeng. Khmer optical character recognition (ocr). 2015.

[8] Bayram Annanurov and Norliza Mohd Noor. Khmer handwritten text recognition with convolution neural networks. *ARPN Journal of Engineering and Applied Sciences*, 13(22):8828–8833, 2018.

[9] Dona Valy, Michel Verleysen, Sophea Chhun, and Jean-Christophe Burie. A new khmer palm leaf manuscript dataset for document analysis and recognition: Sleukrith set. In *Proceedings of the 4th International Workshop on Historical Document Imaging and Processing*, pages 1–6, 2017.

[10] Rina Buoy, Masakazu Iwamura, Sovila Srun, and Koichi Kise. Toward a low-resource non-latin-complete baseline: an exploration of khmer optical character recognition. *IEEE Access*, 11:128044–128060, 2023.

[11] Debabrata Paul and Bidyut Baran Chaudhuri. A blstm network for printed bengali ocr system with high accuracy. *arXiv preprint arXiv:1908.08674*, 2019.

[12] Tayyab Nasir, Muhammad Kamran Malik, and Khurram Shahzad. Mmu-ocr-21: Towards end-to-end urdu text recognition using deep learning. *IEEE Access*, 9:124945–124962, 2021.

[13] Vladimir I Levenshtein et al. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union, 1966.

[14] Jeonghun Baek, Geewook Kim, Junyeop Lee, Sungrae Park, Dongyoon Han, Sangdoo Yun, Seong Joon Oh, and Hwalsuk Lee. What is wrong with scene text recognition model comparisons? dataset and model analysis. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4715–4723, 2019.

[15] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015.