Syllable Analysis Data Augmentation for Khmer Ancient Palm leaf Recognition

Nimol Thuon†, Jun Du†* and Jianshu Zhang‡
† National Engineering Research Center of Speech and Language Information Processing
University of Science and Technology of China, Hefei, Anhui, China
E-mail: tnimol@mail.ustc.edu.cn, jundu@ustc.edu.cn
‡ iFLYTEK Research, Hefei, Anhui, China
E-mail: xysszjs@mail.ustc.edu.cn

***Abstract*—The unique forms and physical conditions of the Khmer palm leaf manuscript recognition system are receiving more attention from researchers. In the state-of-the-art, data augmentation is commonly used for data training; however, grammatical mistakes and data availability in the training process would determine or limit the accuracy rate. The two significant challenges lie in (1) grammar complexity and (2) wording similarity; therefore, this paper presents the Syllable Analysis Data Augmentation (SADA) technique, which aims at boosting the accuracy of the text recognition system for one of Southeast Asia's historical manuscripts from Cambodia. SADA comprises two fundamental modules: (1) formulating a collection of syllables/words to structure glyph patterns and (2) generating patterns from existing data through augmentation techniques and utilizing flexible geometric image transformation to increase similar word/text images. Initially, image collections are established, whereby datasets are interpreted according to the reordered grammatical structures to construct multiple glyph images. Next, we aim at conducting the experiment with a text/word recognition system before regulating attention-based encoder-decoder to enhance the probability of transcriptions of low and high-resolution images. At last, the experiment centers on datasets from various sources, including public datasets from ICFHR 2018 contest and our new augmentation datasets, all of which aim at demonstrating and evaluating the accuracy of the findings.**

I. Introduction

The Khmer palm leaf manuscripts of Cambodia [1] contain a wealth of historical records, culture, and knowledge. Historical manuscripts have documented information about religion, culture, daily lives, art, and education. Regardless, Cambodians seem to have difficulty in reading and writing these scripts thoroughly. As shown in Fig. 1, Southeast Asian researchers [2] collected data from various sources, including museums, cultural organizations, private collections, and religious temples using the digital camera. Unfortunately, datasets are mostly damaged due to physical problems, unstable lighting, aging manuscript quality, and inadequate datasets. In Southeast Asia, collections of palm leaf manuscripts have attracted researchers who are interested in ancient palm leaf analysis including document binarization, text-line segmentation, isolated character classification, and text recognition. Recently, there have been several publications on palm leaf manuscript analysis from Cambodia [3], [4], [5] and Indonesia [6]. In addition to this, the ICHFR 2016 [7] and 2018 [8] contests were examined for their pre-processing and post-processing tasks. Nevertheless, several limitations remain particularly in the pre-processing and text/word recognition steps.

Fig. 1. Samples of Khmer palm leaf manuscripts, including text-line manuscripts, glyphs and text/words.

For machine learning systems, data augmentation [9] is a typical way to increase the amount of data training. To enhance data augmentation, primary and advanced techniques have been implemented successfully in many areas of text image recognition including Chinese [10], Arabic [11], historical Japanese [12], and mathematics recognition systems [13]. Southeast Asian historical languages, on the other hand, have complex grammar and alphabetical components because of the writing and ordering structures. These complexities have also paved ways for newly proposed techniques by researchers worldwide [8]; however, the research does not only neglect the similarities and internal systems of the characters, but also leave out characters' classes and grammatical structures.

Khmer words [14] can be decomposed into different elements of the syllables, each of which can be extracted into different components. Additionally, morphology, phonology, and vocabulary evolved significantly over the past centuries. Moreover, grouping the syllables needs to follow Khmer grammar rules and writing forms. The recent study [4] even mentioned the challenges of lacking datasets and proposed end-to-end frameworks that combine glyph images with straightforward

*corresponding author

writing forms. Consequently, these approaches often struggle to provide a flexible structural analysis of glyphs when combining syllables through ordering images. Furthermore, the study did not focus on generating from existing data as some vowels can stand in front of consonants and some after consonants. Therefore, the Khmer language should investigate specific solutions for dealing with the current challenges.

This study presents Syllable Analysis Data Augmentation (SADA) to boost end-to-end accuracy of an attention-based model in the Khmer palm leaf recognition system. SADA is composed of two components: creating new collections based on glyph patterns and applying flexible geometric data augmentation methods based on syllabic structures. Moreover, we evaluate different training performances using encoder-decoder and attention models. Since the previous study [8] emphasizes the importance of feature extraction for analyzing medium Khmer palm leaf datasets, our study will also highlight pre-processing and feature extraction steps for boosting the accuracy rate.

This paper mainly contributes the following:

1) We introduce SADA data augmentation for Khmer historical documents. To the best of our knowledge, glyph dictionary and eight different syllable forms have been designed based on Khmer grammar structures.
2) We evaluate end-to-end system on different benchmarks, by which we equipped and examined the encoder as the feature extractor on different architecture of CNNs and decoder as GRU with attention model.
3) We access the most effective end-to-end system using SADA datasets to determine the performance and accuracy rate of our techniques.
4) We further analyze strengths and weaknesses with and without data augmentations to see how deep learning approaches are involved in recognition system.

II. Khmer Language and Datasets

Until the present, Cambodian is defined by the Khmer language, which belongs to Austroasiatic languages known as a member of Mon-Khmer languages [15]. The Khmer language consists of more than 28 vowels and 33 consonants as the primary, by which each of the consonants carries its own subscripts. Apart from the main features, the writing structures [14] are also formed by memorable characters, including diacritical marks and independent characters. Moreover, the enormous glyphs indicate Khmer as the complex on clustering syllable and writing form.

SleukRith set [2] is a dataset constructed on the captured image of Khmer palm leaf manuscripts, containing types of glyphs, words, and text-line documents. A glyph is a short component of a syllable consisting of a single or few consonants and vowels. In this study, glyphs images match a syllable based on correct writing forms, followed by flexible geometric transformations. According to Khmer's writing form, most of the primary Khmer syllables are consonants and vowels, especially in the Khmer palm leaf dataset.

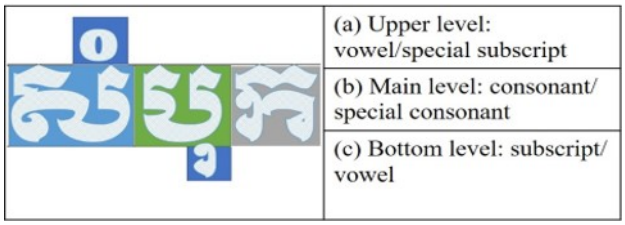


Fig. 2. Khmer word levels. a) Upper level for vowel and individual subscript; b) Main level for central consonant and singular consonant; c) Bottom level for subscript and particular vowel.

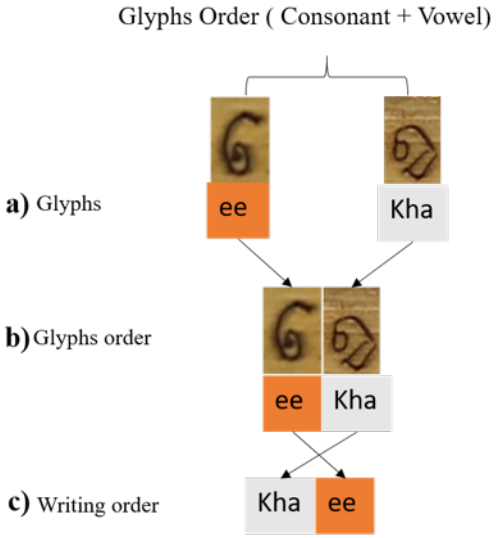


Fig. 3. A group of glyph images samples. a) Two input images represent vowel "ee" and consonant "Kha"; b) For glyphs order: some vowels can be in front, and some vowels stay behind consonants; c) For writing order: consonants always stay in front of vowels.

Despite the quantity, another interesting supplement to the complexity of Khmer language lies in its formation behavior or the ways glyph is grouped to form a word. Two significant indications shall be taken into consideration. Firstly, as shown in Fig. 2, the characters of the Khmer language are made up of three different levels: upper level, main level, and bottom level. Furthermore, the figure indicates that the main level (b) may include consonants or particular consonants, whereas the upper level and bottom level may contain a vowel, special subscripts, or subscripts as the upper level (a) and bottom level (c). The combinations of the consonant with particular subscript, subscript, diacritical mark, or vowel are the main elements, although the characters can only be formed by understanding the order and position of each type of glyphs and grammatical rules. Furthermore, syllable has one or two consonant clusters. In contrast, if Khmer words begin with consonants and vowels, the vowels located infront or behind depending on the type of vowels, followed by the consonants, as illustrated in Fig. 3.

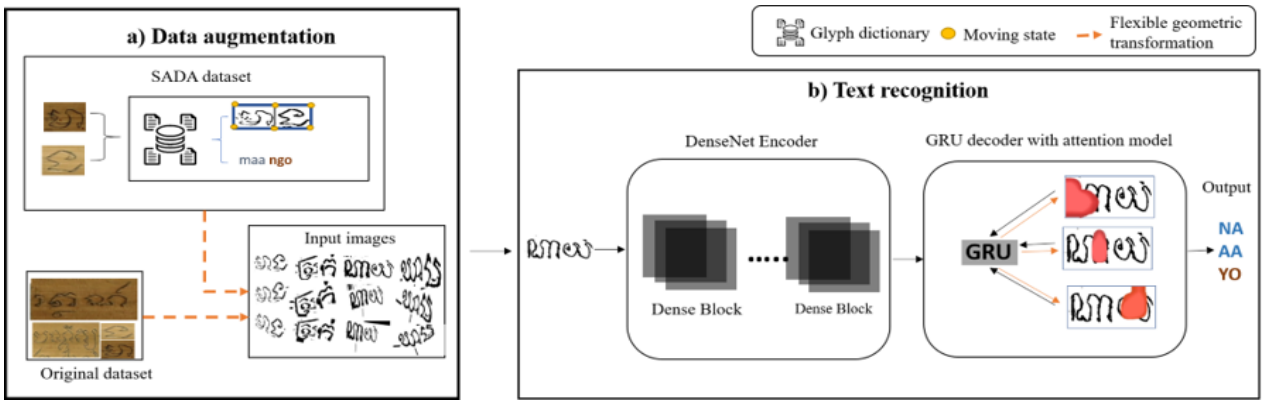


Fig. 4. The diagram aims at providing an overview of SADA architecture. In the first phase, we construct new syllable datasets by pairing multiple isolated images based on the glyph dictionary. The flexible data augmentation will then generate this augmentation based on random moving states on the newly generated datasets as well as existing datasets. Finally, we perform encoder-decoder text recognition to inspect the results based on the different sizes of the data.

III. Overall Frameworks

Figure 4 shows an overview of the architecture, including data augmentation and text recognition. In this paper, we propose a data augmentation strategy for improving Khmer palm leaf recognition by using isolated glyph patterns. Firstly, the syllables reconstruct by combining random glyphs based on grammar structures. Furthermore, the glyph dictionary and syllable forms have been presented. Finally, we evaluate our datasets based on deep learning approaches, namely encoder-decoder with the attention-based mechanism.

A. Syllable Data Augmentation

In this study, we aim to generate new syllable datasets by extracting data from Khmer isolated datasets [2]. This section is classified upon the ground of four steps: (1) establish glyph dictionary that contain consonants and vowels; (2) combine glyph images and thresholding; (3) adopt flexible geometric transformation; (4) construct a new label list for the transliteration training.

1) Glyph dictionary: According to Khmer's writing form, most of the primary Khmer syllables are consonants and vowels. Generally, consonants and vowels are combined to form a syllable based on writing forms. In this case, the vowel should be on the right side of the consonant, which differs from the syllable translation in Latin where consonant locates in front of vowels. As shown in Fig. 5, the grammatical structures are presented, incorporating eight common sentence structures in Khmer language. To decompose Khmer words into different class types, we create a dictionary of each image identified by label and style. The groups of consonant and grammatical principles of vowel are demonstrated to ease the process of validating new data generation as follow:

- VC : Vowel + Consonant
- CC : Consonant + Consonant
- VV : Vowel + Vowel
- CV : Consonant + Vowel
- CCC : Consonant + Consonant + Consonant

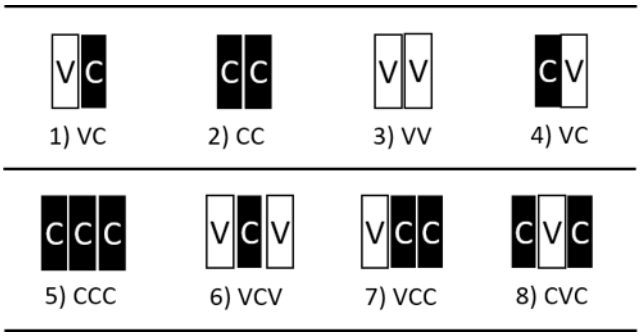


Fig. 5. Illustration of Khmer common grammar structures.

- VCV : Vowel + Consonant + Vowel
- VCC : Vowel + Consonant + Consonant
- CVC : Consonant + Vowel + Consonant

2) Combining glyph images and thresholding: In this step, two different glyph images are randomly chosen from the Khmer isolated dataset and put together before employing thresholding to enhance the quality of the images. The combination of the glyphs is conducted in a simple procedure where initially input images in the dictionary need to match the labels and type of the glyphs to determine consonants or vowels. Following the identification process, the glyph will go through grammatical structure possibilities in the files and combine horizontally in the same size to formulate accurate wordings. Furthermore, we recommend resizing and arranging multiple images at once to avoid multiple resizing and deteriorating the quality of the images. We can also deploy basic operations like horizontal concatenation to create and reshape images.

After concatenation, we apply thresholding to enhance low-quality and degraded documents through noise eradication and to make the system more realistic. Furthermore, to address the challenges of low quality, we adopt the updated version of Sauvola (ISauvola) [16] by extracting an initial map to increase the probability of text regions.

3) Flexible geometric transformation: We generate augmented data in the final stage using flexible geometric transformation. Recently, Luo *et al.* [17] proposed smart data augmentations for handwritten text images. It contained random geometric transformation approaches. This technical step first established a batch of points on the image representing fiducial points. The movement of points provides a "harder" training sample, which is then used to generate a new image using moving similarity transformation [18]. The augmentation module additionally receives a random movement state to create a random augmented image. We may then employ fiducial points with a higher degree of customization. The bottom and top of the image are defined by $2(P+1)$ fiducial points. However, to generate efficient augmented data, we are required to set a fit image moving path, as shown in Fig. 6. In here, we selected a single image path for the moving points. Unlike the handwritten dataset, Khmer syllable or training datasets contained short words and incomplete sentences, which is why image path P = 1 is generated better than another image path. Our next step is to enhance images by randomly generate fiducial points c inside the radius R. We generate an augmented image by giving x as a point in the image and then transforming x as follows:

$$H(x) = (x - p_*)\, L + c_*, \tag{1}$$

where the linear transformation matrix L is formed from a scalar, the weighted centroids of the initialized fiducial point are p and c. In regards to point x, weight w_i has the form below:

$$p_* = \frac{\sum_{i=1}^{2(P+1)} w_i p_i}{\sum_{i=1}^{2(P+1)} w_i}, c_* = \frac{\sum_{i=1}^{2(P+1)} w_i c_i}{\sum_{i=1}^{2(P+1)} w_i} \tag{2}$$

$$w_i = \frac{1}{|p_i - x|^{2\alpha}} \tag{3}$$

The weight w_i increases as x approaches p_i. The movement of the nearest fiducial point mainly determines the value of x. In the case that w_i is bounded and $x = p_i$. if we set $*$ to 1. Using the simplest possible transformation, H_x, we can maximize following:

$$\sum_{i=1}^{2(P+1)} w_i \left| H_x\left(p_i\right) - c_i \right|^2 \tag{4}$$

By randomly moving the fiducial points, we can distort an image as in Fig. 6.

4) Create a new label list for transliteration training: After generating a new Khmer syllable, we need to pair it with a new label. A new label of an image containing type and label is created by combining labels following our grammar points. For instance, as illustrated in Fig. 3, if we input one consonant "Kha" and one vowel "ee", our system will initially identify their label and type before combining them into one group. In this case, "Kha" and "ee" follow the grammatical order VC and grammatical transliteration label CV. The new dictionary will be created following these structures = (img, label, class) and our dictionary type will illustrate ("Khaee.jpg", "Khaee", "CV").

Fig. 6. Flexible data augmentation. Comparing the results among three different image patches ($0 < P < 4$).

B. Word/text recognition

The overall system of an end-to-end framework is displayed in Fig. 4 (b), which contains an encoder and decoder with attention mechanisms, whereby the input image is initially encoded into high-level representations. At this point, the decoder utilizes the context vector to construct the output sequence based on each word.

1) Encoder: Dense Convolutional Network (DenseNet) [19] is selected to extract features from input images as it performs better than other encoders. Particularly, instead of removing all features after a fully connected layer, the encoder discards the fully connected layer and the softmax layer. In addition, feature extractors, such as DenseNet, can aid in learning the correct alignment between local regions and feature maps. In this case, the decoder can then selectively focus on a specific portion of the images based on its visual characteristics. The preceding feature maps are taken as inputs for each layer. When deconstructing received all feature maps as:

$$x_k = H_l\left([x_0, x_1, x_2 \cdots x_{k-1}]\right) \tag{5}$$

where $x_0, x_1, x_2 \cdots x_{k-1}$ is the ordered features for layer $0, \cdots\cdots, k-1$ and H_l is known to be a single tensor. The Batch-Normalization (BN) [20], ReLU [21] and a convolution process are performed in consecutive operations. The convolutional operations are conducted with BN and an average pooling layer in the transaction block accordingly. An encoder generates variable-length grids of L pieces ($L = H \times W$), which are used to produce high-level visual representations of $H \times W \times D$. The array also contains elements that identify local regions of the images as D-dimensional annotation vectors:

$$\mathbf{A} = \{\mathrm{a}_1, \ldots, \mathrm{a}_L\}, \mathrm{a}_n \in \mathbb{R}^D \tag{6}$$

2) Decoder: The decoder is used to transcribe sequences to Latin output because of its abilities to solve vanishing and exploding gradient problems; therefore, we employ Gated Recurrent Units (GRUs) as the decoder [13]. $\mathbf{Y}$ represents the output sequence of encoding characters.

$$\mathbf{Y} = \{\mathrm{y}_1, \ldots, \mathrm{y}_K\}, \mathrm{y}_n \in \mathbb{R}^G \tag{7}$$

where G stands for the total number of words, including the sequence of words, and K denotes the total length. GRU uses context vectors c_t, hidden states s_t, and target words y_t to determine the forecast term's GRU probability. The probability of each predicted word can be determined:

$$\mathrm{P}\left(\mathrm{y}_t \mid \mathrm{y}_{t-1}, \mathbf{X}\right) = g\left(\mathrm{F}_o h\left(\mathrm{E}\ \mathrm{y}_{t-1} + \mathrm{F}_a \mathrm{s}_t + \mathrm{F}_c \mathrm{c}_t\right)\right) \quad (8)$$

where $\mathbf{X}$ indicates input images, the softmax activation function is g, h is the maxout activation function, F_o, F_a, F_c and E corresponds to the embedding matrix and GRU decoding dimensions. In order to determine the hidden state s_t, the decoder employs two unidirectional GRU layers, which are as follows:

$$\hat{\mathbf{s}}_t = \mathrm{GRU}\left(\mathbf{y}_{t-1}, \mathbf{s}_{t-1}\right) \quad (9)$$

$$\mathbf{c}_t = f_{\mathrm{c}}\left(\hat{\mathrm{s}}_t, \mathbf{A}\right) \quad (10)$$

$$\mathbf{s}_t = \mathrm{GRU}\left(\mathrm{c}_t, \hat{\mathrm{s}}_t\right) \quad (11)$$

Similarly, s_{t-1} indicates the previous hidden state, s_t indicates the predicted current hidden state of the GRU, and f_c corresponds to the coverage-based spatial attention model.

3) Coverage based attention model: Using coverage as a measure, we can determine whether a certain section of the original image has been translated. In the previous step, we compute all the coverage vectors using prior attention probabilities, instead of retention probabilities. In our view, the sum of prior attention probabilities indicates better alignment. Therefore, we constructed an attention model based on the coverage vector using the following equations:

$$\mathbf{M} = \mathbf{N} * \sum_{l=1}^{t-1} \alpha_l \quad (12)$$

$$\alpha_{ti} = \frac{\exp\left(e_{ti}\right)}{\sum_{k=1}^{L} \exp\left(e_{tk}\right)} \quad (13)$$

$$e_{ti} = v_a^{\mathrm{T}} \tanh\left(\mathbf{W}_a \mathbf{h}_{\mathrm{t}-1} + \mathbf{U}_a \mathbf{a}_i + \mathbf{U}_f \mathbf{f}_i\right) \quad (14)$$

where e_{ti} is the hidden state prediction s_t. The coverage vector $\mathbf{f}_i$ of the annotation vector $\mathbf{a}_i$. We add all previous attention probabilities starting from a zero-valued coverage vector. At time step t, α_{ti} represents the spatial attention coefficient of $\mathbf{a}_i$. Assuming n' indicates the attention dimension, and z represents the output channels of the convolution function $\mathbf{N}$; then $\nu_{\mathrm{a}} \in \mathbb{R}^{n'}$, $\mathbf{W}_a \in \mathbb{R}^{n' \times n}$, $\mathbf{U}_a \in \mathbb{R}^{n' \times D}$ and $\mathbf{U}_f \in \mathbb{R}^{n' \times z}$.

IV. Experimental setups

The experiments aim to analyze the generated datasets to validate the data augmentation approach's efficacy and robustness for Khmer historical documents.

A. Evaluation Scenarios

In this experiment, we examine the performance of our entire frameworks before and after the utilization of augmentation strategies. Respectively, CER (Character Error Rate) and WER (Word Error Rate), are selected to validate the performance trend with different sizes of augmented. Moreover, we also show the improvements and the limitations of attention visualization of each representative. Thus, we divide our evaluations to two scenarios following:

1) Scenario 1: We evaluate different encoder architectures including VGG, ResNet and DenseNet on binary/greyscale input images.
2) Scenario 2: We select the best recognizer system to evaluate the effect of our data augmentation techniques on different scale of datasets.

B. Training procedure

In order to reduce some noises, ISauvola [16] was used to convert all of the datasets to binary images. We began by implementing DenseNet encoder. The main branch of DenseNet has been implemented with three dense blocks. Before proceeding to the first layer, the input images are first convoluted to 7×7 with 48 channels, 2×2 of max-pooling layer is then applied. The bottleneck layers are used to improve computational efficiency, so a 3×3 convolution is preceded by a 1×1 convolution. The growth rate for each block is k = 24, and the depth is D = 32. Each block consists of 16 convolution layers of 1×1 and 16 layers of 3×3. After the convolution layer, the ReLU activation layer is applied after the batch normalization layer. In the decoder, GRU is used on a single layer, by setting to 256 for dimension m of the embedding and dimension n. The attention model is adjusted by setting the annotation dimension D, and the size of the convolution kernel for computing the coverage vector is 7×7. Accordingly, the adadelta technique [22] is applied with gradient clipping to optimize the model.

C. Dataset

The results are generated based on different types of datasets as evident in Table I. These new datasets will be categorized into training, validation, and testing datasets. See details of data visualizations in Fig. 7. In this experiment, we use five different datasets for training the models. The five datasets are presented as follows:

- ICFHR: Publicly datasets from ICFHR 2018 contest [8].
- SADA-X: New data augmentation that we generate from existing training of ICFHR 2018.
- SADA-Y-1: Dataset contains only the combination of multiple glyphs without generating additional data augmentation.
- SADA-Y-2: New data augmentation that generates from SADA-Y-1.
- SADA-Mixed: Dataset mixed from all of the above datasets.

TABLE I
THE OVERVIEW OF THE TOTAL OF TRAINING, TESTING AND VALIDATION DATASETS.

Dataset	Training	Testing	Validation
ICFHR 2018 [8]	16,333	7,791	850
SADA-X	56,000		
SADA-Y-1	36,600		
SADA-Y-2	183,000		
SADA-Mixed	291,933		

Fig. 7. Data visualization of the newly generated data presenting: (a) examples of how the new data augmented from word/text dataset using flexible geometric transformations; and (b) samples of the new collections generated from the isolated character datasets using SADA techniques.

V. RESULTS AND ANALYSIS

Khmer historical documents present a variety of challenges, making it difficult for a recognition system to cover them all. Therefore, it is necessary to analyze datasets and approaches simultaneously. Consequently, this section presents the results of different end-to-end frameworks with SADA datasets, including the effectiveness and limitations of Khmer historical recognition analysis.

A. Performance trend analysis

The results of **scenario 1 & 2** are displayed in Table II and Table III, respectively. In **scenario 1**, we compared our approaches to two of the previous winners from the ICFHR 2018 contest [8]. Likewise, we used the same testing, training, and validating data configurations as the ICFHR contest. Among them, our DenseNet-GRU with binary datasets has the best performance. Afterward, we selected DenseNet-GRU to investigate the performance of **scenario 2**. We discovered

TABLE II
SCENARIO 1. THE COMPARISONS OF **S1-S5**, WHERE **S1-S2** WERE REPORTED BY [8] AND **S3-S5** WERE EVALUATED USING BINARY IMAGES.

System	Encoder	Data	CER	WER
S1 [8]	VGG	Greyscale	4.51	-
S2 [8]	DenseNet	Greyscale	**3.38**	-
S3	VGG	Binary	3.67	17.93
S4	ResNet	Binary	3.51	16.11
S5	**DenseNet**	**Binary**	**3.18**	**14.93**

TABLE III
SCENARIO 2. THE EVALUATIONS RESULTS OF THE EXPERIMENT ARE BASED ON DENSENET-GRU (S5) USING SADA DATASETS.

Dataset	CER	WER
S5 + ICFHR	3.18	14.93
S5 + SADA-Y-1	16.05	39.08
S5 + SADA-Y-2	13.36	25.04
S5 + SADA-X	**3.05**	**13.69**
S5 + SADA-Mixed	**2.45**	**11.07**

some interesting information through our data augmentation methods. In particular, DenseNet-GRU performed better on both SADA-X and SADA-Mixed datasets, indicating that our newly generated datasets were beneficial for boosting Khmer palm leaf recognition. However, if we only trained on SADA-Y-1 and SADA-Y-2 without the original ICFHR datasets, the results would be worse due to the mismatch between augmented data and realistic data testing.

B. Data augmentation analysis

In this section, the performances of the training are analyzed based on different size of training data. As mentioned above, SADA-Mixed are consisted of the original datasets and the newly generated datasets; therefore, we begin by investigating the training results at different size of data from the original size (∽16K) to 18 times increase (∽291K) accordingly. In

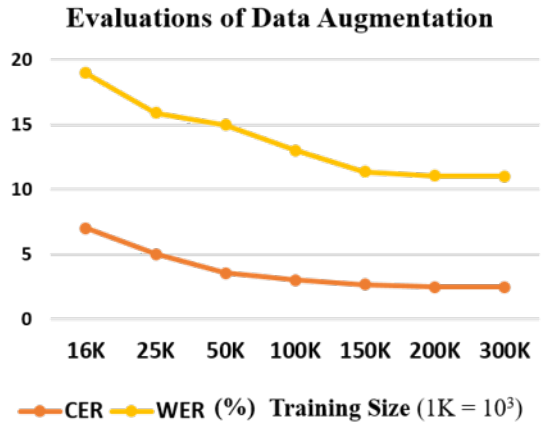


Fig. 8. Relationship between error rates and data quantity. The results are analyzed based on the size of training data using SADA-Mixed datasets.

Fig. 9. The example of prediction with attention visualizations. The outputs were correctly predicted by using (a) DenseNet-GRU with SADA techniques and (b) DenseNet-GRU without any pre-processing techniques.

particular, the data size are randomly selected in sequence order from minimum to maximum for performance evaluation. As shown in Fig. 8, the error rates of this recognition system are affected by the quantity of data deployed in the training, by which more error rates are reduced on larger datasets. Data augmentation is undeniably essential in boosting performance analysis. However, the scale of the data alone cannot determine higher accuracy rates. In this aspect, we should also put more emphasis on the efficacy and quality of data during the training phase, as only the combination of quantity and quality can boost the accuracy rates for analyzing complex grammatical structures like the Khmer language.

C. Error analysis and attention visualization

The decoder with the attention model demonstrated an impressive performance at the transcription stage. Visualizing attention probabilities helps us understand how SADA strategies recognize Khmer word structures. Additionally, we further analyze the error examples using attention visualization. For instance, we present side-by-side comparisons of greyscale and binary images in Fig. 9. Our first example shows misclassified characters in Fig. 9 (b) using the DenseNet-GRU using attention-based recognition without any pre-processing steps. The results revealed some misrecognition and word order arrangement struggles based on Khmer grammatical writing. In contrast, as shown in Fig. 9 (a), training with SADA strategies aligns the word order structures of characters with the Khmer writing system. Additionally, larger datasets also led to more accurate predictions. Further details are provided in Fig. 10, which shows the comparison of sample results from different SADA datasets. As a result of mixing the original datasets with the additional dataset (SADA-Mixed), some misrecognition of characters and performances were fairly well corrected. In this way, SADA strategies showed their potential for boosting the accuracy of end-to-end frameworks for complex grammar and ancient documents such as palm leaf manuscripts. In spite of this, limitations are considered to be the problem when recognizing long words. Therefore, Khmer palm leaf analysis still has room for improvements in future studies.

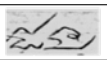
Datasets			
1) SADA-X	nongo ✗	daaba ✗	<blank> ✗
2) SADA-Mixed	dango ✓	baaba ✓	roee ✗
3) Ground-truth	dango ✓	baaba ✓	rotoeeah ✓

Fig. 10. Analysis of the prediction results, highlighting on the effectiveness and limitations of the SADA datasets.

VI. Conclusion

In this paper, we investigated the quality and complex grammar structures of Khmer palm leaf documents. Specifically, we have presented a data augmentation strategy (SADA) based on a glyph dictionary and flexible geometric transformation for Khmer historical palm leaf recognition. In addition, the standard Khmer grammar structures and Khmer glyph dictionary are designed to generate new collections from isolated character patterns. On the basis of the ICFHR dataset and our newly generated datasets, we assessed the strengths and weaknesses of various attention-based recognition systems. Furthermore, the experiments demonstrated that our data augmentation strategy significantly enhanced the recognition system's performance in correcting grammatical and word order errors. In future works, we intend to improve recognition performance through the extension of feature extractions and post-processing step.

Acknowledgment

This project was supported by the Chinese Academy of Science and the World Academic Science President's Fellowship.

References

[1] M. W. A. Kesiman et al., "Benchmarking of document image analysis tasks for palm leaf manuscripts from southeast asia," *Journal of Imaging,* vol. 4, no. 2, p. 43, 2018.

[2] D. Valy, M. Verleysen, S. Chhun, and J.-C. 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,* 2017, pp. 1–6.

[3] D. Valy, M. Verleysen, S. Chhun, and J.-C. Burie, "Character and text recognition of khmer historical palm leaf manuscripts," in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR),* 2018, pp. 13–18.

[4] D. Valy, M. Verleysen, and S. Chhun, "Data Augmentation and Text Recognition on Khmer Historical Manuscripts," in *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR), 2020,* pp. 73–78.

[5] D. Valy, M. Verleysen, and S. Chhun, "Text Recognition on Khmer Historical Documents using Glyph Class Map Generation with Encoder-Decoder Model.," in *ICPRAM,* 2019, pp. 749–756.

[6] M. W. Kesiman et al., "Southeast Asian palm leaf manuscript images: a review of handwritten text line segmentation methods and new challenges," *Journal of Electronic Imaging,* vol. 26, no. 1, p. 011011, 2016.

[7] J.-C. Burie et al., "ICFHR2016 competition on the analysis of handwritten text in images of balinese palm leaf manuscripts," in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR),* 2016, pp. 596–601.

[8] M. W. A. Kesiman et al., "ICFHR 2018 competition on document image analysis tasks for southeast asian palm leaf manuscripts," in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR),* 2018, pp. 483–488.
[9] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data,* vol. 6, no. 1, pp. 1–48, 2019.
[10] X. Shen and R. Messina, "A method of synthesizing handwritten chinese images for data augmentation," in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR),* 2016, pp. 114–119.
[11] M. Eltay, A. Zidouri, I. Ahmad, and Y. Elarian, "Improving Handwritten Arabic Text Recognition Using an Adaptive Data-Augmentation Algorithm," in *International Conference on Document Analysis and Recognition,* 2021, pp. 322–335.
[12] N. T. Ly, C. T. Nguyen, and M. Nakagawa, "Training an end-to-end model for offline handwritten Japanese text recognition by generated synthetic patterns," in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR),* 2018, pp. 74–79.
[13] J. Zhang et al., "Watch, attend and parse: An end-to-end neural network based approach to handwritten mathematical expression recognition," *Pattern Recognition,* vol. 71, pp. 196–206, 2017.
[14] J. M. Jacob, "The structure of the word in Old Khmer," *Bulletin of the school of Oriental and African Studies,* vol. 23, no. 2, pp. 351–368, 1960.
[15] M. J. Alves and others, *A grammar of Pacoh: a Mon-Khmer language of the central highlands of Vietnam.* Pacific Linguistics, Research School of Pacific and Asian Studies, the …, 2006.
[16] Z. Hadjadj, A. Meziane, Y. Cherfa, M. Cheriet, and I. Setitra, "ISauvola: Improved Sauvola's algorithm for document image binarization," in *International Conference on Image Analysis and Recognition,* 2016, pp. 737–745.
[17] C. Luo, Y. Zhu, L. Jin, and Y. Wang, "Learn to augment: Joint data augmentation and network optimization for text recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 2020, pp. 13746–13755.
[18] S. Schaefer, T. McPhail, and J. Warren, "Image deformation using moving least squares," in *ACM SIGGRAPH 2006 Papers,* 2006, pp. 533–540
[19] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition,* 2017, pp. 4700–4708.
[20] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning,* 2015, pp. 448–456.
[21] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375,* 2018.
[22] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701,* 2012.