WildKhmerST: A Comprehensive Dataset and Benchmark for Khmer Scene Text Detection and Recognition in the Wild

Vannkinh Nom, Saly Keo, Souhail Bakkali, Muhammad Muzzamil Luqman, Mickaël Coustaty, Marçal Rossinyol, Jean-Marc Ogier

WildKhmerST: A Comprehensive Dataset and Benchmark for Khmer Scene Text Detection and Recognition in the Wild

Vannkinh Nom[1,2*], Saly Keo[1,2*], Souhail Bakkali[1], Muhammad Muzzamil Luqman[1], Mickaël Coustaty[1], Marçal Rossinyol[3], and Jean-Marc Ogier[1]

[1] La Rochelle University, Laboratoire Informatique Image Interaction (L3i)
[2] Cambodia Academy of Digital Technology (CADT)
[3] AllRead Machine Learning Technologies, Spain
{vannkinh.nom, keo.saly, souhail.bakkali, muhammad_muzzamil.luqman, mickael.coustaty, jean-marc.ogier}@univ-lr.fr; marcal@allread.ai

Abstract. This study presents a large-scale dataset of Khmer scene text images captured in real-world environments. Khmer, the official language of Cambodia, is spoken by approximately 17 million people. While Optical Character Recognition (OCR) systems have achieved remarkable success in Roman (Latin) script languages such as English, Khmer script poses unique challenges due to its intricate structure, absence of clear word boundaries, and highly diverse character shapes and sizes. A significant limitation in Khmer OCR research has been the scarcity of high-quality training data, particularly for deep learning-based models, which require extensive datasets to achieve robust performance. To address these challenges, we introduce a newly constructed dataset of Khmer scene text, comprising 29,601 annotated text lines from 10,000 unique images. This dataset is highly diverse and challenging, encompassing artistic text, blurred text, low-light conditions, curved text, text in complex backgrounds, and occluded text. Each text line is annotated with polygonal bounding box coordinates and line-level transcriptions, alongside attributes describing background complexity, character appearance, and text style. To establish a foundational benchmark for future research in Khmer OCR, we provide baseline results for Khmer text detection and recognition. Additionally, we propose a robust evaluation metric tailored for Khmer OCR, enabling precise assessment of CER and WER while accounting for the unique characteristics of the Khmer script.

Keywords: WildKhmerST Dataset · Khmer Script · Low-resource languages · Scene-Text Detection and Recognition

1 Introduction

In recent years, extracting text from images has attracted considerable interest due to its wide-ranging applications, such as document analysis [14,32], image-

* Nom and Keo contributed equally to this work.

Fig. 1: An example of a complex Khmer word that combines all parts, including consonant (red), vowel (green), subscript (blue), and diacritics (orange).

based translation [9,44], product image retrieval [17,18], visual geo-location [24,37], and license plate recognition [8,42]. Thanks to advanced deep learning models, researchers have made great progress in detecting and recognizing text in various languages, including Latin, Chinese, and Arabic scripts [19,20,25,36]. In the case of Khmer, deep learning has been applied in several areas: the authors of [39] worked on text recognition in historical manuscripts, in [5] the authors focused on recognizing Khmer text in synthetic datasets, the authors of [35] addressed on Khmer printed text, and in [28], the authors explored text detection and recognition in Khmer scene text. Unlike Latin-based scripts, Khmer script constructs words in a distinct manner. A distinguishing characteristic of Khmer is that consonants change shape depending on their spatial arrangement within a word. This includes consonants and subscript, where multiple consonants can merge into new forms—often referred to as subscripts or low consonants—positioned beneath the main consonant. This structural combination plays a crucial role in determining consonant sounds. Additionally, vowels in Khmer are highly flexible in placement, occurring before, after, above, or below consonants. In certain cases, vowels can also merge to form entirely new vowel sounds [39]. Fig. 1 illustrates a Khmer word that combines various script components, including consonants, subscript, vowels, and diacritics. These difficulties, along with the fact that many non-Latin languages are low-resource. As a result, reading Khmer text in the wild needs more well-annotated training samples. Unfortunately, existing benchmarks [28] struggle to meet these needs due to the high cost of data collection, with only 1,544 images available. To address this issue, we developed a new, large-scale Khmer scene text dateset called WildKhmerST, which includes 29,601 Khmer text lines from 10,000 unique sample images. The dataset is diverse and challenging, containing planar text, raised text, low-light text, distant text, and partially occluded text. Alongside the dataset, we also provide a benchmark for Khmer scene text detection and recognition tasks. However, to accurately assess model performance on this diverse dataset, reliable evaluation metrics are necessary. Since many languages such as Latin, Chinese, and Arabic scripts use the levenshtein distance [25,36], which measures errors by counting insertions, deletions, and substitutions between the predicted and ground truth text. Khmer script presents unique challenges due to its complex structure and encoding variations. As shown in Fig. 2, on the Character cluster, some words may look the same but have different character arrangements. This makes it difficult to compare them without checking their actual character order. To address the challenges of Khmer character clusters, where the same word can be represented by different sequences, we propose the Khmer OCR Evaluation Metrics

(a) Ambiguous consonants	(1) ក vs គ vs ត	(2) ប vs ម	(3) ជ vs ផ
(b) Ambiguous subscripts	(1) ◌្ក vs ◌្ខ vs ◌្ត vs ◌្ទ vs ◌្ព vs ◌្ភ	(2) ◌្ជ vs ◌្ឈ	(3) ◌្ញ vs ◌្ឋ vs ◌្ឌ
(c) Ambiguous dependent vowels	(1) ◌ិ vs ◌ី vs ◌ឹ vs ◌ឺ	(2) ◌ៃ vs ◌ៃ	(3) ◌ៅ+◌ី vs ◌ើ
(d) Ambiguous Independent vowels	(1) ឧ vs ឱ vs ឩ vs ឪ vs ឫ vs ឬ	(2) ឮ vs ឯ	(3) ឭ vs ឮ vs ឭ
(e) Ligatures	(1) = ប + ◌ា	(2) ស៊ី = ស + ◌ិ + ◌ី	(3) ប៉ង = ប + ◌ុ + ◌ំ + ង
(f) Character cluster	គន្ត្រី	(1) ត + ន + ◌្ត + ◌្រ + ◌ី (2) ត + ន + ◌្រ + ◌្ត + ◌ី (3) ត + ន + ◌្រ + ◌ ◌ + ◌្ត	
	ហើ	(1) ហ + ◌ៅ + ◌ី (2) ហ + ◌ើ	

Fig. 2: Examples of Khmer character clusters, ligatures, and ambiguous characters, with different variations. For base consonants (red), subscripts (blue), diacritics (orange), dependent vowels (green), and independent vowels (pink).

named KHCWER[1]. These metrics are specifically designed to evaluate CER and WER in the context of the Khmer language. The key component of this method is Khmer word normalization process and word segmentation, which applied the Khmer word segmentation of khmercut[2] to split both the predicted and ground-truth texts and Khmer Syllable Reordering Search[3] for Khmer text normalization; we adapted only the parts related to reordering each syllable while ensuring that the original structure of the text remains intact.

In this research, we made two key contributions. First, we introduced the Khmer Scene Text Images dataset, which consists of 10,000 images featuring real-world Khmer text. Additionally, we provided benchmark model performance for Khmer text detection and recognition. Second, we proposed the Khmer OCR Evaluation Metrics, specifically designed to assess CER and WER in the context of the Khmer language. These metrics incorporate robust normalization and word segmentation techniques to ensure consistent and accurate OCR performance evaluation. They address the complexities of the Khmer script, including subscript ordering, vowel reordering, and diacritic handling.

2 Related Work

2.1 Datasets of Text in Natural Images

Detecting and recognizing scene text in natural images with unconstrained environments remains a challenging task in computer vision. The ability to accurately read text in such settings can greatly benefit various real-world applications. To

[1] https://github.com/keosaly/KHCWER.git
[2] https://github.com/seanghay/khmercut
[3] https://github.com/Trey314159/KhmerSyllableReordering

further improve the understanding of text in natural scenes, [40] introduced the COCO-Text dataset, specifically designed for scene text detection and recognition. This dataset is built upon the Microsoft COCO dataset, which provides annotations for common objects in their natural contexts. Datasets of text in scene images can be categorized into two main types: real images captured from real-world environments [23,40] and synthetic text images [11,15]. For example, the authors in [26] introduced a new benchmark dataset for research purposes, consisting of over 600,000 labeled digit images cropped from Street View Images. In [45] the authors presented a large dataset of Chinese text in natural images, named Chinese Text in the Wild (CTW). The dataset contains 32,285 images with 1,018,402 Chinese characters. Since the Khmer language has significantly fewer publicly available datasets compared to Latin scripts, researchers have worked on generating synthetic and real-world datasets to support Khmer OCR and scene text recognition. One such effort is by [5], who synthetically generated datasets for both document OCR and scene text recognition using the Khmer corpus and fonts available in Tesseract. These synthetic datasets aim to compensate for the lack of real annotated data by creating large-scale training samples for Khmer text recognition models. Another notable contribution in Khmer scene text images is the KhmerST dataset introduced by [28], specifically designed to advance computer vision research focused on Khmer script recognition. This dataset comprises 1,544 images, which are categorized into indoor and outdoor scenes. This dataset consists of a diverse collection of images captured from various public places across Cambodia, including streets, signboards, supermarkets, and commercial establishments, all containing Khmer text.

2.2 Text Detection and Recognition

In recent years, text detection and recognition have seen significant advancements driven by both traditional and deep learning-based methods. Classical approaches rely on hand-crafted features such as stroke width and edge detection, while modern deep learning techniques automatically learn discriminative features through convolutional and recurrent networks. Text detection identifies and localizes text regions in an image, serving as a crucial preprocessing step for recognition. Traditional methods rely on hand-crafted features like edge detection, stroke width transform (SWT) [13], and maximally stable extremal regions (MSER) [27]. These approaches often struggle with complex backgrounds and varying text styles. Recent deep learning methods, such as CTPN [38] and EAST [46], leverage convolutional neural networks (CNNs) to detect text regions more effectively by learning hierarchical features. More recent approaches, such as CPN [43] and ATTR [47], enhance detection performance further by incorporating complementary proposal mechanisms and Transformer-based multi-scale feature aggregation, respectively. The You Only Look Once (YOLO) model [31] detects objects in a single pass. Unlike traditional multi-stage methods, YOLO examines the entire image at once during training and testing. This enables it to efficiently capture contextual information, leading to faster, more accurate detections. Building on this, [28] proposed various YOLO models to detect

Khmer text in scene images. By leveraging YOLO's single-pass processing, their approach aimed to improve localization efficiency and adaptability to Khmer script complexities in real-world environments. In our case, we treat the text line as an object and use different YOLO versions to detect it effectively. Text recognition converts detected text regions into machine-readable text, typically using sequence modeling techniques. Early approaches relied on optical character recognition (OCR) techniques based on template matching and feature extraction [16]. With deep learning, CNNs combined with recurrent neural networks like LSTMs became popular for text sequence modeling [33]. [34] introduced the convolutional recurrent neural network (CRNN), integrating CNNs for feature extraction, RNNs for sequence modeling, and Connectionist Temporal Classification (CTC) loss for transcription. More recent advancements, including the deep text recognition benchmark [2] and ViTSTR [1], have further improved recognition accuracy by capturing long-range dependencies. Additionally, models such as PARSeq [3] enhance performance by using permuted autoregressive decoding, which effectively handles diverse and irregular text sequences. TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models [22] introduced a novel approach using a Transformer encoder-decoder, allowing it to learn contextual representations from large-scale text datasets. By eliminating recurrent components, TrOCR achieves high recognition accuracy while maintaining computational efficiency. In our approach, we utilize the CRNN, deep text recognition, and TrOCR models to evaluate the effectiveness of these methods for text line recognition tasks.

3 WildKhmerST Dataset

In this section, we introduce the comprehensive dataset for Khmer scene text in natural environments, known as "WildKhmerST". As shown in Fig. 3, the dataset covers a variety of real-world complexities, such as varying lighting conditions, occlusions, cluttered backgrounds, and different text orientations, sizes, and fonts. We will explain how the images are annotated, how the dataset is split into training, validation, and provide statistics on the dataset. The dataset and evaluation metric will be publicly accessible[4].

3.1 Dataset and Annotations

The annotation process was conducted using the VGG Image Annotator (VIA) [12], a powerful tool for creating detailed and accurate image annotations. VIA enables the definition of regions within images using polygon coordinates, allowing for the precise outlining of complex shapes by specifying vertices along the x and y axes. In each image, all instances of Khmer text lines are annotated, while text lines in English and other languages are excluded from the annotations. The annotation workflow is illustrated in Fig. 4, where a polygon-based bounding box is initially drawn around each sentence of Khmer text. The annotation

[4] http://l3i-share.univ-lr.fr/2025WildKhmerST/

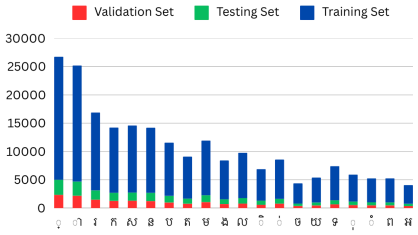
Fig. 3: Examples of WildKhmerST dataset illustrate diverse text layouts, including curved, varying lighting conditions, occlusions, cluttered backgrounds, and different text orientations, sizes, and fonts.

Fig. 4: The examples of polygon-based bounding box annotations for Khmer text in real-World scenes.

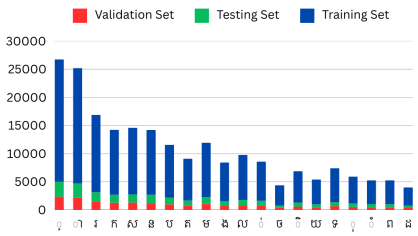
data for each image, including detailed annotations, is organized in JSON format. By accommodating text rotations and contours, the polygon-based method significantly enhances recognition accuracy, particularly in cases where text deviates from standard horizontal alignment. The JSON entries also store essential metadata, such as image filenames and dimensions, enhancing the dataset's effectiveness for training and evaluating deep learning models specialized in Khmer script detection and recognition.

3.2 Statistics

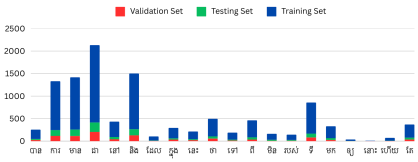
WildKhmerST is a dataset that includes 10,000 images containing 12,108 unique words collected from diverse real-world text sources such as street signs, banners, advertisements, store boards, posters, and product labels. To better understand our dataset, we conducted an analysis of the most frequently occurring characters and words. In Fig. 5a, we present the distribution of Khmer character instances across the training set, validation set, and testing set in WildKhmerST. This analysis is based on the 20 most frequently occurring Khmer words identified in



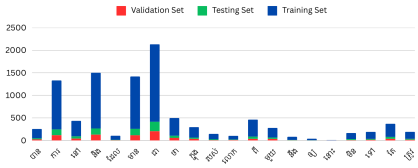
(a) The top 20 most frequently Khmer character based on SEAlang library.



(b) The top 20 most frequently Khmer character based on Kheng dictionary



(c) The top 20 most frequently Khmer words based on SEAlang library.



(d) The top 20 most frequently Khmer words based on Kheng dictionary

Fig. 5: Analysis of the most frequently occurring Khmer characters and words in WildKhmerST Dataset, with blue representing the training set, red for the validation set, and green for the testing set.

the SEAlang library[5], which is a comprehensive resource offering language reference materials for Southeast Asian languages. Similarly, in Fig. 5b, we display the distribution of Khmer character instances in WildKhmerST, but this time based on the 20 most frequently searched words in the online Khmer language dictionary known as Kheng[6].

For a deeper exploration of word frequency in the dataset, Fig. 5c shows the distribution of Khmer word instances in the training set, validation set, and testing set of WildKhmerST, based on the 20 most frequently occurring Khmer words in the SEAlang library. Likewise, Fig. 5d, we extend this analysis by displaying the distribution of Khmer word instances in WildKhmerST using the 20 most frequently searched Khmer words from the Kheng dictionary. From the statistics, we can see that WildKhmerST contains a significantly larger number of samples for the most frequent characters and words based on the SEAlang and Kheng dictionary. However, we still found some less common words in the dataset, such as (ឲ្យ) appearing 39 times and (ទោះ) appearing only 10 times.

[5] http://sealang.net/project/list/
[6] https://kheng.info/frequencies/

4 Evaluation Metrics

The widely used metrics to evaluate the effectiveness of deep learning techniques in identifying text sequences from input images is the levenshtein distance [25]. Both CER and WER are based on the levenshtein distance, which calculates the ratio of unrecognized characters in the predicted result compared to the total characters in the ground truth. This typically includes insertions, deletions, and substitutions. A lower value for these metrics indicates higher accuracy of the model. These metrics are chosen because they are commonly used to assess the accuracy of OCR [6,7,25,28]. While CER is effective for many languages, Khmer script requires deeper consideration due to its complex written structure, as shown in Fig. 2. Challenges arise from ambiguous consonants, subscripts, vowels, ligatures, character clusters, and other unique script features. To address this issue, the character normalization scheme proposed in the previous research of [4] standardizes these sequences into a canonical form. This normalization is achieved by decomposing each character cluster into smaller units (such as subscripts and vowels), applying rule-based corrections at the unit level, and then recombining the corrected units into a normalized cluster. This approach was applied in the character normalization process of [5].

To evaluate the performance of our model, we first applied the Khmer Syllable Reordering Search for Khmer text normalization; we adapted only the parts related to reordering each syllable while ensuring that the original structure of the text remains intact. The reordering process involves several key steps: first, we remove zero-width elements and identify the base character, which is either a consonant or an independent vowel. Additionally, specific vowel combinations are repaired, such as replacing (ᨠ + ᩮ) with (ᨦᩮ), and (ᨠ + ᩁ) with (ᨣᩁ). The subscript consonants are reordered, ensuring that "ro" (᩠ᩁ) always appears last, swapping positions if coeng + ro comes before coeng + another consonant. Finally, the syllable components as shown in Fig. 2, are joined in a structured order: base character + other register shifters + robat + coeng chunks + dependent vowels + non-spacing diacritics + spacing diacritics, maintaining proper Khmer script formation. Throughout this process, we avoided de-duplicating groups to ensure that if a chunk appears multiple times in sequence, it is preserved instead of being reduced to a single instance. Similarly to the previous research [6,7,28], levenshtein distance is used to compute the CER for each word to find the ratio of unrecognized characters over the total number of characters in the target words. Here, S is number of substitutions, I is number of inserts, D is number of deletes, and N is number of ground-truth characters. Since the Khmer language does not have any rule of explicit word boundaries[5], we use a word segmentation algorithm of khmercut to split both the predicted and ground-truth texts before calculating the WER.

$$CER = \frac{S + I + D}{N}; \quad WER = \frac{\text{Number of incorrect words}}{\text{Total number of words in ground truth}} \tag{1}$$

Table 1: Comparison of CER and WER calculations across different evaluation metrics: Jiwer, TorchMetrics, and KHCWER.

Ground-Truth	Ground-Truth Seg.	Instance Seg.	Jiwer		TorchMetrics		KHCWER	
			CER	WER	CER	WER	CER	WER
ខ្ញុំស្តាប់តន្ត្រី	['ខ្ញុំ', 'ស្តាប់', 'តន្ត្រី']	['ខ្ញុំ', 'ស្តាប់', 'តន្ត្រី']	0.17	1.0	0.17	1.0	0	0
ចាបហើលើមេឃ	['ចាប', 'ហើលើ', 'មេឃ']	['ចាប', 'ហើលើ', 'មេឃ']	0.18	1.0	0.1	1.0	0	0
តន្ត្រីស្រុកស្រែ	['តន្ត្រី', 'ស្រុក', 'ស្រែ']	['តន្ត្រី', 'ស្រុក', 'ស្រែ']	0.13	1.0	0.13	1.0	0	0
ស្តាប់ភ្លេងខ្មែរ	['ស្តាប់', 'ភ្លេង', 'ខ្មែរ']	['ស្តាប់', 'ភ្លេង', 'ខ្មែរ']	0.06	1.0	0.06	1.0	0	0
ខោរហែក	['ខោ', 'រហែក']	['ខោ', 'រហែក']	0.17	1.0	0.17	1.0	0	0

A comparison of CER and WER calculations across three different metrics, Jiwer[7], TorchMetrics[8], and our custom KHCWER metric, is provided in Table 1. We compare ground-truth text with instances that appear visually identical but differ in vowel placement order. Jiwer and TorchMetrics produce identical results for both CER and WER, while KHCWER shows slightly different outcomes. Notably, KHCWER consistently yields lower CER and WER values, suggesting a more lenient or refined approach to error calculation.

5 Benchmark Model and Performance

5.1 Text Line Detection

Detecting text lines in Khmer script presents unique challenges due to the script's complex character shapes, varying text orientations, and the diverse backgrounds often found in natural scene images. To address these challenges, we leverage the YOLO models, which are well-suited for real-time object detection tasks. Previous research has demonstrated the effectiveness of YOLO for text detection tasks, with [45] proposing YOLOv2 for Chinese character detection and line detection in scene images and [28] utilizing different versions of YOLO to detect Khmer text lines in natural images. Motivated by these studies, we specifically utilize YOLOv8, YOLOv10, and YOLOv11 to detect text lines in scene images, aiming to evaluate and compare their performance in handling the unique complexities of Khmer script. YOLOv8 introduces improved accuracy and flexibility with enhanced anchor-free detection and advanced data augmentation techniques. Meanwhile, YOLOv10 and YOLOv11 further push the limits by optimizing the efficiency of the model, reducing computational overhead, and improving the detection performance in dense and small text regions. The advantages of using YOLO models include their speed, scalability, and ability to handle complex scenes with varying text orientations, sizes, and backgrounds. Additionally, their lightweight nature allows deployment on resource-constrained devices, making them ideal for real-world scene text detection. We resize the image to 800x800 pixels for the input image pair, with the text format coordinates

[7] https://pypi.org/project/jiwer
[8] https://lightning.ai/docs/torchmetrics/stable

Table 2: Detection result performance across different YOLO models, including YOLOv8, YOLOv10, and YOLOv11.

Model	Params (M)	Precision	Recall	mAP50	mAP50-90
YOLOv8s	11	0.827	0.803	0.88	0.581
YOLOv10s	8	0.831	0.797	0.877	0.572
YOLOv11s	9	**0.840**	**0.809**	**0.889**	**0.583**

(x_center, y_center, width, height) and a class ID, where '0' denotes Khmer text line as the object of interest. The model performance is evaluated using metrics such as precision, mAP50, and mAP50-95. mAP50 calculates the overlap between predicted and actual bounding boxes, considering a match valid if the overlap is 50% or more. On the other hand, mAP50-95 offers a more comprehensive accuracy evaluation by assessing performance across various overlap thresholds ranging from 50% to 95%. We conducted the experiment using Wild-KhmerST dataset for training and evaluated the models on the KhmerST dataset [28]. As shown in Table 5, YOLOv11s achieved the highest precision value of 0.840, outperforming YOLOv8s at 0.827 and YOLOv10s at 0.831. Additionally, YOLOv11s demonstrated superior recall at 0.809 and mAP50 at 0.889, making it the most accurate model for text detection tasks. In contrast, YOLOv10s recorded the lowest recall of 0.797, while YOLOv8s achieved a recall of 0.803.

In terms of computational efficiency, YOLOv10s is the most lightweight model, with only 8 million parameters, but lags slightly in performance compared to YOLOv11s. Despite its compact size, it has a higher computational cost than YOLOv8s and YOLOv11s. YOLOv11s, with 9 million parameters, balances complexity and efficiency, delivering superior accuracy. Its optimized architecture excels at detecting small and dense text regions, common in natural scene images. Its ability to capture intricate details in text with varying fonts, sizes, and orientations makes it particularly effective for real-world applications. On the other hand, YOLOv10s, while efficient in terms of parameters, sacrifices some accuracy, as reflected in its lower recall and mAP50 scores of 0.797 and 0.877, respectively. YOLOv8s remains a competitive choice, offering a good balance of precision, recall, and efficiency. The detection results of the YOLO models are illustrated in Fig.6, with YOLOv8s shown in Fig.6a, YOLOv10s in Fig.6b, and YOLOv11s in Fig.6c. These models are highly effective at detecting text lines, even identifying small text lines or line numbers not present in the ground truth test set [28].

Overall, YOLOv11s emerges as the best-performing model for text detection tasks, particularly in complex scenes with diverse text orientations and backgrounds. Its high precision, recall, and mAP50 scores, combined with its computational efficiency, make it the ideal choice for real-world applications. While YOLOv10s offers lightweight efficiency and YOLOv8s provides a balanced performance-runtime trade-off, YOLOv11s stands out as the most reliable and accurate model overall.

(a) Detection results from YOLOv8s. For each image, the text line detection areas are highlighted with their corresponding bounding boxes.

(b) Detection results from YOLOv10s. For each image, the text line detection areas are highlighted with their corresponding bounding boxes.

(c) Detection results from YOLOv11s. For each image, the text line detection areas are highlighted with their corresponding bounding boxes.

Fig. 6: Examples of text line detection using YOLO models. Ground-truth bounding boxes are shown in green, and model predictions in yellow. The models can even detect small text lines that are not annotated in the ground truth.

5.2 Text Line Recognition

Given a cropped text line region from the ground truth, the primary goal of the recognition task is to accurately extract textual information. To evaluate the effectiveness of various approaches, we tested the performance of several state-of-the-art models, including a CRNN [33], a deep text recognition benchmark model [2], and TrOCR [22]. We conducted three different experiments and, to assess the evaluation metrics, calculated the CER and WER using Equation 1.

In the first experiment, we evaluated several models on the WildKhmerST dataset using an 80:10:10 train-validation-test split. We began with a CRNN model (VGG + CTC), which achieved a CER of 0.85% and WER of 1.39%, establishing a baseline but struggling with complex text. To improve accuracy,

Table 3: Recognition performance across different models and experimental setups: Experiment 1 involves training, validation, and testing on our proposed WildKhmerST dataset. Experiment 2 trains on WildKhmerST and tests on the KhmerST [28]. Experiment 3 is fully trained, validated, and tested on KhmerST.

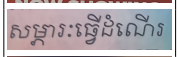
Model	Backbone	Decoder	Experiment 1 KHCWER(%)		Experiment 2 KHCWER(%)		Experiment 3 KHCWER(%)	
			CER	WER	CER	WER	CER	WER
CRNN	VGG	CTC	0.85	1.39	0.74	1.43	0.94	1.21
TVBC	VGG	CTC	0.32	0.74	0.29	0.66	**0.86**	**1.16**
TVBA	VGG	Attention-base	0.51	0.99	0.21	0.50	0.93	1.17
TRBC	ResNet	CTC	0.29	0.70	0.24	0.56	0.96	1.45
TRBA	ResNet	Attention-base	0.33	0.72	0.19	0.46	0.90	1.10
TrOCR_1	-	Tr.Decoder	**0.13**	**0.30**	0.19	0.37	0.94	1.11
TrOCR_2	-	Tr.Decoder	0.16	0.34	**0.18**	**0.35**	0.95	1.12

we tested deeper architectures. Among VGG-based models, TVBC (TPS-VGG-BiLSTM-CTC) outperformed TVBA (TPS-VGG-BiLSTM-Attention) with a CER of 0.32% and WER of 0.74%. Among ResNet-based models, TRBC (TPS-ResNet-BiLSTM-CTC) achieved the best results with CER of 0.29% and WER of 0.70%. We also evaluated TrOCR, a Transformer-based OCR model, using NLLB and XMLRoberta tokenizers. TrOCR1 (NLLB) achieved CER of 0.13%, WER of 0.30%; TrOCR2 (XMLRoberta) reached CER of 0.16%, WER of 0.34%. TrOCR outperformed all previous models. Finally, we selected the top three—TRBC, TrOCR1, and TrOCR2—to extract text from random validation images. Compared to the ground truth, the models accurately extracted the text in most cases, with only a few errors highlighted in red, as illustrated in Table 4.

In the second experiment, we evaluated the generalization capabilities of the models by training on our proposed WildKhmerST dataset and testing on the KhmerST dataset [28], introducing a domain shift to assess adaptability to unseen data. The CRNN model achieved a CER of 0.74% and WER of 1.43%, showing slight improvement over Experiment 1 but still lagging behind other models. Among VGG-based models, TVBA outperformed TVBC with a CER of 0.21% and WER of 0.50%, highlighting the robustness of attention-based decoders to domain shifts. For ResNet-based models, TRBA performed best, achieving a CER of 0.19% and WER of 0.46%, reinforcing the benefits of combining ResNet with attention mechanisms for cross-dataset generalization. Transformer-based TrOCR models also showed strong performance: TrOCR1 achieved CER of 0.19% and WER of 0.37%, while TrOCR2 slightly outperformed it with CER of 0.18% and WER of 0.35%, reaffirming the effectiveness of Transformer architectures in handling domain shifts.

In the third experiment, we fully trained and tested the models on the KhmerST dataset [28] using the same data split as in Experiment 1, with an 80:10:10 ratio for training, validation, and testing, respectively. As shown in Table 3, despite training and testing on the same dataset, the models yielded high

Table 4: Comparison of recognition results for the top 3 models using TRBC, TrOCR1, and TrOCR2 in Experiment 1. Red text highlights erroneous predictions compared to the ground truth, indicating where each model fails.

Instance	Ground-Truth	TRBC	TrOCR1	TrOCR2
	នៅពេលដែល	នៅពេលដែល	នៅពេលដែល	នៅពេលដែល
	អាចដូរបណ្តុលបាន	អាចដូរបណ្តាលបាន	អាចដូរបណ្តាលបាន	អាចដូបទល្ងាចបាន
	គ្រឹះស្ថានឯកជន	គ្រឹះស្ថានឯកជន	គ្រឹះស្ថានឯកជន	គ្រឹះស្ថានឯកជន
	ទទួលសញ្ញាបត្រ	ទទួលសញ្ញាបត្រ	ទទួលសញ្ញាបត្រ	ទទួលសញ្ញាបត្រ
	កំពុងចាក់បញ្ចាំង	កំពុងចាក់បញ្ចាំង	កំពុងចាក់បញ្ចាំង	កំពុងចាក់បញ្ចាំង
	សម្ភារៈធ្វើដំណើរ	សម្ភារៈធ្វើដំណើរ	សម្ភារៈធ្វើដំណើរ	សម្ភារៈធ្វើដំណើរការ
	ទូកចោរសមុទ្រ	ទូកបោរសមុទ្រ	ទូកលារសមុទ	ទូករចនាសមុទ

CER and WER, indicating difficulty in learning and generalizing from the data. This highlights potential limitations in dataset size. The results suggest that the KhmerST dataset, being smaller than WildKhmerST, lacks sufficient data for learning robust representations, resulting in higher error rates. All experiment results are summarized in Table 3.

Overall, the experiments demonstrate the effectiveness of different architectures for text recognition tasks under varying conditions. In Experiment 1, where models were trained and tested on the proposed WildKhmerST dataset, TrOCR emerged as the top performer, showcasing the superiority of Transformer-based models. In Experiment 2, which introduced a domain shift by training on WildKhmerST and testing on the KhmerST dataset, TRBA and the TrOCR models demonstrated strong generalization capabilities. These results highlight the importance of attention mechanisms and Transformer-based architectures in handling domain shifts, whereas traditional models like CRNN struggled to generalize effectively. However, in Experiment 3, we observed that the model produced very high CER and WER, indicating poor performance. This issue arose due to the limited size of the KhmerST dataset used for training.

5.3 Discussions and Limitations

Detection. Despite achieving strong performance metrics with a precision of 0.840, recall of 0.809, mAP50 of 0.889, and mAP50-90 of 0.583 when trained on WildKhmerST and tested on KhmerST datasets, the YOLO model exhibits notable limitations in detecting Khmer text in scene images. One key challenge is its reduced accuracy when processing small text instances and images with poor lighting conditions, as illustrated in Fig. 7, where the model inconsistently detects text, missing some instances entirely. This limitation highlights the need for more advanced detection strategies or model enhancements that can better handle low-resolution and low-contrast text in complex real-world scenes.

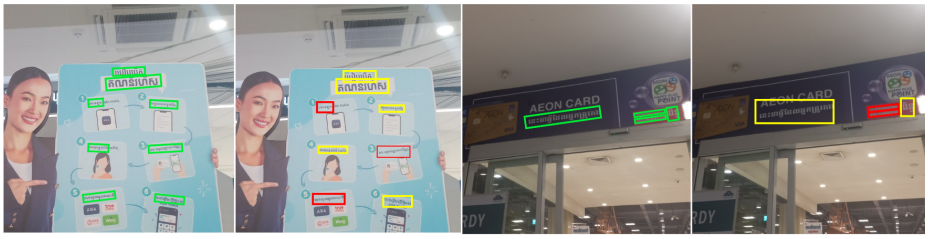


Fig. 7: Example of detection failure cases. Images (1) and (3) show ground truth annotations, with text lines highlighted in green, while (2) and (4) display model detections, where predicted boxes are marked in yellow and missed ones in red.

Recognition. Based on the results of Experiment 2, our proposed dataset, consisting of 29,601 text lines from 10,000 unique images, provides a robust foundation for training Khmer OCR models. The impact of domain shift is evident, as models trained on WildKhmerST were tested on the unseen KhmerST dataset. While CRNN and CTC-based models experienced notable performance drops, attention-based models like TRBA and Transformer-based models demonstrated superior generalization. Their ability to extract text from unseen data highlights the effectiveness of attention mechanisms and Transformer architectures in handling variations in text style, noise, and font diversity, making them more suitable for real-world Khmer OCR, as shown in Table 4. However, in Experiment 1, where models were trained on 80% of WildKhmerST, the TRBA model showed a performance drop. In contrast, Transformer-based models remained robust, with TrOCR1 achieving the lowest CER and WER. This suggests that Transformers are more resilient to domain shifts and reduced training data compared to CRNN, CTC-based, and Attention-based models. Moreover, the word-level analysis in Table 5 highlights the advantages of Transformer-based models in recognizing the 10 most frequent Khmer words in Experiment 1. Traditional models like CRNN and TVBC had lower accuracy, especially on common words, whereas TrOCR1 and TrOCR2 consistently achieved the highest accuracy, reflecting their strong ability to capture contextual dependencies and character interactions. Furthermore, Experiment 3 illustrated the challenges of training on a limited dataset, as models trained solely on KhmerST struggled with high CER and WER. This reinforces the importance of dataset scale and diversity in improving model robustness and generalization. Despite promising results, the models struggle with recognizing complex samples, particularly freestyle writing and irregular horizontal lines, which are underrepresented in our dataset. These cases introduce high variability in character shapes, spacing, and orientation, making it difficult to extract consistent patterns. As shown in Fig. 8, freestyle writing often includes uneven strokes, overlapping characters, and inconsistent baselines, while horizontal lines may suffer from skewed alignment. This limitation underscores the need for more diverse data and advanced preprocessing to improve robustness against complex text structures.

Table 5: Top-1 accuracy (%) of the 10 most frequent Khmer words across different text recognition models in Experiment 1.

Model	និង	លក់	មាន	គ្រប់	លេខ	ផ្ទះ	ភ្នំពេញ	ទឹក	ផ្លូវ	សង្កាត់
CRNN	11.12	0	25.50	0	23.80	0	0	20.30	0	0
TVBC	45.30	57.90	57.90	40.28	50.85	54.45	35.72	71.22	30.96	10.72
TVBA	36.12	62.75	57.02	48.62	47.46	60.61	28.58	55.80	26.2	7.2
TRBC	50.0	64,71	67.11	48.62	59.33	57.58	38.1	73.92	73.92	14.29
TRBA	61.12	75.50	72.81	58.34	66.95	62.13	45.24	75.37	47.62	32.15
TrOCR1	**92.86**	**95.59**	**90.79**	**95.84**	**99.16**	**93.94**	**90.48**	**92.76**	89.29	**92.86**
TrOCR2	**92.86**	94.61	90.36	**95.84**	94.07	90.91	**90.48**	**92.76**	**91.67**	**92.86**

Fig. 8: Example of recognition failure cases, including freestyle writing and horizontal text lines, which contribute to recognition difficulties.

6 Conclusion

In conclusion, this research addresses the challenges of Khmer text recognition by introducing a large-scale, diverse dataset of 29,601 text lines from 10,000 unique sample images, annotated with polygonal bounding boxes and line-level transcriptions. The dataset captures real-world complexities, including artistic text, blurry text, low-light text, curved text, text with complex backgrounds, and occluded text, providing a robust foundation for training and evaluation. We also propose a Khmer-specific OCR evaluation metric that incorporates normalization and word segmentation to handle the script's unique complexities, such as subscript ordering, vowel reordering, and diacritic handling. Through extensive benchmarking, we demonstrate the effectiveness of state-of-the-art models by identifying effective models for scene text detection and recognition tasks and analyzing their strengths and weaknesses. These contributions establish a strong baseline for future research and pave the way for advancements in Khmer OCR, particularly in addressing challenges like ambiguous characters, complex layouts, and low-resolution text. We believe that our dataset and evaluation framework will greatly stimulate future work in Khmer text detection and recognition, enabling further innovations in this field.

In future work, we plan to develop a scene text generation model specifically for the Khmer language. Additionally, we will explore the design of a joint or end-to-end model tailored for Khmer text, integrating detection and recognition into a unified framework.

References

1. Atienza, R. (2021, September). Vision transformer for fast and efficient scene text recognition. In International conference on document analysis and recognition (pp. 319-334). Cham: Springer International Publishing.
2. Baek, J., Kim, G., Lee, J., Park, S., Han, D., Yun, S., ... & Lee, H. (2019). What is wrong with scene text recognition model comparisons? dataset and model analysis. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 4715-4723).
3. Bautista, D., & Atienza, R. (2022, October). Scene text recognition with permuted autoregressive sequence models. In European conference on computer vision (pp. 178-196). Cham: Springer Nature Switzerland.
4. Buoy, R., Taing, N., & Chenda, S. (2021). Khmer Word Search: Challenges, Solutions, and Semantic-Aware Search. arXiv preprint arXiv:2112.08918.
5. Buoy, R., Iwamura, M., Srun, S., & Kise, K. (2023). Toward a low-resource non-latin-complete baseline: an exploration of khmer optical character recognition. IEEE Access, 11, 128044-128060.
6. Buoy, R., Taing, N., Chenda, S., & Kor, S. (2022). 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.
7. Buoy, R., Kor, S., & Taing, N. (2021). An End-to-End Khmer Optical Character Recognition using Sequence-to-Sequence with Attention. ArXiv, abs/2106.10875.
8. Chang, S. L., Chen, L. S., Chung, Y. C., & Chen, S. W. (2004). Automatic license plate recognition. IEEE transactions on intelligent transportation systems, 5(1), 42-53.
9. Cho, W., Choi, S., Park, D. K., Shin, I., & Choo, J. (2019). Image-to-image translation via group-wise deep whitening-and-coloring transformation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10639-10647).
10. Davis, R. H., & Lyall, J. (1986). Recognition of handwritten characters—a review. Image and vision computing, 4(4), 208-218.
11. de Campos, T. E., Babu, B. R., & Varma, M. (2009, February). Character recognition in natural images. In International conference on computer vision theory and applications (Vol. 1, pp. 273-280). SCITEPRESS.
12. Dutta, A., & Zisserman, A. (2019, October). The VIA annotation software for images, audio and video. In Proceedings of the 27th ACM international conference on multimedia (pp. 2276-2279).
13. Epshtein, B., Ofek, E., & Wexler, Y. (2010, June). Detecting text in natural scenes with stroke width transform. In 2010 IEEE computer society conference on computer vision and pattern recognition (pp. 2963-2970). IEEE.
14. Gilani, A., Qasim, S. R., Malik, I., & Shafait, F. (2017, November). Table detection using deep learning. In 2017 14th IAPR international conference on document analysis and recognition (ICDAR) (Vol. 1, pp. 771-776). IEEE.
15. Jaderberg, M., Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Synthetic data and artificial neural networks for natural scene text recognition. arXiv preprint arXiv:1406.2227
16. Jain, A. K., & Bhattacharjee, S. (1992). Text segmentation using Gabor filters for automatic document processing. Machine vision and applications, 5(3), 169-184.
17. Jang, Y. K., & Cho, N. I. (2021). Self-supervised product quantization for deep unsupervised image retrieval. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 12085-12094).

18. Jo, J., Lee, S., Lee, C., Lee, D., & Lim, H. (2020). Development of fashion product retrieval and recommendations model based on deep learning. Electronics, 9(3), 508.
19. Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., i Bigorda, L. G., Mestre, S. R., ... & De Las Heras, L. P. (2013, August). ICDAR 2013 robust reading competition. In 2013 12th international conference on document analysis and recognition (pp. 1484-1493). IEEE.
20. Karatzas, D., Gomez-Bigorda, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., ... & Valveny, E. (2015, August). ICDAR 2015 competition on robust reading. In 2015 13th international conference on document analysis and recognition (ICDAR) (pp. 1156-1160). IEEE.
21. Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. Proceedings of the Soviet physics doklady.
22. Li, M., Lv, T., Chen, J., Cui, L., Lu, Y., Florencio, D., ... & Wei, F. (2023, June). Trocr: Transformer-based optical character recognition with pre-trained models. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 37, No. 11, pp. 13094-13102).
23. Lucas, S. M., Panaretos, A., Sosa, L., Tang, A., Wong, S., Young, R., ... & Lin, X. (2005). ICDAR 2003 robust reading competitions: entries, results, and future directions. International Journal of Document Analysis and Recognition (IJDAR), 7, 105-122.
24. Muller-Budack, E., Pustu-Iren, K., & Ewerth, R. (2018). Geolocation estimation of photos using a hierarchical model and scene classification. In Proceedings of the European conference on computer vision (ECCV) (pp. 563-579).
25. Nasir, T., Malik, M. K., & Shahzad, K. (2021). Mmu-ocr-21: Towards end-to-end urdu text recognition using deep learning. IEEE Access, 9, 124945-124962.
26. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., & Ng, A. Y. (2011, December). Reading digits in natural images with unsupervised feature learning. In NIPS workshop on deep learning and unsupervised feature learning (Vol. 2011, No. 2, p. 4).
27. Neumann, L., & Matas, J. (2012, June). Real-time scene text localization and recognition. In 2012 IEEE conference on computer vision and pattern recognition (pp. 3538-3545). IEEE.
28. Nom, V., Bakkali, S., Luqman, M. M., Coustaty, M., & Ogier, J. M. (2024). KhmerST: A Low-Resource Khmer Scene Text Detection and Recognition Benchmark. In Proceedings of the Asian Conference on Computer Vision (pp. 1777-1792).
29. Paul, D., & Chaudhuri, B. B. (2019). A BLSTM network for printed Bengali OCR system with high accuracy. arXiv preprint arXiv:1908.08674.
30. Rawls, S., Cao, H., Sabir, E., & Natarajan, P. (2017, April). Combining deep learning and language modeling for segmentation-free OCR from raw pixels. In 2017 1st international workshop on Arabic script analysis and recognition (ASAR) (pp. 119-123). IEEE.
31. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).
32. Schreiber, S., Agne, S., Wolf, I., Dengel, A., & Ahmed, S. (2017, November). Deepdesrt: Deep learning for detection and structure recognition of tables in document images. In 2017 14th IAPR international conference on document analysis and recognition (ICDAR) (Vol. 1, pp. 1162-1167). IEEE.
33. Shi, B., Wang, X., Lyu, P., Yao, C., & Bai, X. (2016). Robust scene text recognition with automatic rectification. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4168-4176).

34. Shi, B., Bai, X., & Yao, C. (2016). An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE transactions on pattern analysis and machine intelligence, 39(11), 2298-2304.
35. Sok, P., & Taing, N. (2014, December). 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 (pp. 1-9). IEEE.
36. Sun, Y., Liu, J., Liu, W., Han, J., Ding, E., & Liu, J. (2019). Chinese street view text: Large-scale chinese text reading with partially supervised learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 9086-9095).
37. Tian, Y., Chen, C., & Shah, M. (2017). Cross-view image matching for geo-localization in urban environments. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3608-3616).
38. Tian, Z., Huang, W., He, T., He, P., & Qiao, Y. (2016). Detecting text in natural image with connectionist text proposal network. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14 (pp. 56-72). Springer International Publishing.
39. Valy, D., Verleysen, M., & Chhun, S. (2020, September). Data augmentation and text recognition on Khmer historical manuscripts. In 2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR) (pp. 73-78). IEEE.
40. Veit, A., Matera, T., Neumann, L., Matas, J., & Belongie, S. (2016). Coco-text: Dataset and benchmark for text detection and recognition in natural images. arXiv preprint arXiv:1601.07140.
41. Wei, T. C., Sheikh, U. U., & Ab Rahman, A. A. H. (2018, March). Improved optical character recognition with deep neural network. In 2018 IEEE 14th international colloquium on signal processing & its applications (CSPA) (pp. 245-249). IEEE.
42. Weihong, W., & Jiaoyang, T. (2020). Research on license plate recognition algorithms based on deep learning in complex environment. IEEE Access, 8, 91661-91675.
43. Wu, L., Tian, S., Wang, Y., & Xiong, P. (2024, March). CPN: complementary proposal network for unconstrained text detection. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 38, No. 6, pp. 6057-6065).
44. Yi, Z., Zhang, H., Tan, P., & Gong, M. (2017). Dualgan: Unsupervised dual learning for image-to-image translation. In Proceedings of the IEEE international conference on computer vision (pp. 2849-2857).
45. Yuan, T. L., Zhu, Z., Xu, K., Li, C. J., Mu, T. J., & Hu, S. M. (2019). A large chinese text dataset in the wild. Journal of Computer Science and Technology, 34, 509-521.
46. Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., & Liang, J. (2017). East: an efficient and accurate scene text detector. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 5551-5560).
47. Zhou, Z., Du, X., Zheng, Y., & Jin, C. (2022). Aggregated text transformer for scene text detection. arXiv preprint arXiv:2211.13984.