KhmerST: A Low-Resource Khmer Scene Text Detection and Recognition Benchmark

Vannkinh Nom[1,2], Souhail Bakkali[1], Muhammad Muzzamil Luqman[1], Mickaël Coustaty[1], and Jean-Marc Ogier[1]

[1] La Rochelle University, Laboratoire Informatique Image Interaction (L3i)
[2] Cambodia Academy of Digital Technology
{vannkinh.nom, souhail.bakkali, muhammad_muzzamil.luqman, mickael.coustaty, jean-marc.ogier}@univ-lr.fr

Abstract. Developing effective scene text detection and recognition models hinges on extensive training data, which can be both laborious and costly to obtain, especially for low-resourced languages. Conventional methods tailored for Latin characters often falter with non-Latin scripts due to challenges like character stacking, diacritics, and variable character widths without clear word boundaries. In this paper, we introduce the first Khmer scene-text dataset, featuring 1,544 expert-annotated images, including 997 indoor and 547 outdoor scenes. This diverse dataset includes flat text, raised text, poorly illuminated text, distant and partially obscured text. Annotations provide line-level text and polygonal bounding box coordinates for each scene. The benchmark includes baseline models for scene-text detection and recognition tasks, providing a robust starting point for future research endeavors. The KhmerST dataset is publicly accessible [1].

Keywords: Khmer script · KhmerST dataset · Scene-Text Detection and Recognition

1 Introduction

Automatic scene-text detection and recognition (STDR) in natural scenes are critical tasks in computer vision, with applications ranging from word spotting [5], text-line detection [28], character-level and/or word-level recognition [1, 28]. This problem has been extensively studied and is divided into two scenarios of varying difficulties: text detection and recognition in natural scenes, which presents significant challenges. In natural scene images, characters can vary greatly in appearance due to differences in style, font, resolution, and illumination. Additionally, text may be partially obscured, distorted, or set against complex backgrounds, complicating detection and recognition. The situation is further complicated by high intra-class variability (*i.e.* differences within the same character) and low inter-class variability (*i.e.* differences between different

[1] https://gitlab.com/vannkinhnom123/khmerst

Consonants	1st Series															2nd Series																					
Base	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							
Subscript	◌្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							
Codepoint	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F	90	91	92	93	94	95	96	97	98	99	9A	9B	9C							

Vowels																																				
Dependent	◌ា	◌ិ	◌ី	◌ឹ	◌ឺ	◌ុ	◌ូ	◌ួ	◌ើ	◌ឿ	◌ៀ	◌េ	◌ែ	◌ៃ	◌ោ	◌ៅ	◌ុំ	◌ំ	◌ាំ	◌ះ	◌ុះ	◌ៈ	◌ោះ														
Codepoint	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF	C0	C1	C2	C3	C4	C5	BB C6	C6	B6 C6	C7	BB C7	C1 C7	C4 C7														
Independent	តំ	ឣ	ឤ	ឥ	ឦ	ឧ	ឩ	ឪ	ឫ	ឬ	ឭ	ឮ	ឯ	ឰ	ឱ	ឲ	ឳ																				
Codepoint	A5	A6	A7	A9	AA	AB	AC	AD	AE	AF	B0	B1	B2	B3																							

Diacritics	◌ំ	◌ិ	◌ៈ	◌ុ	◌ូ	◌ួ	◌ើ	◌ឿ	◌ៀ	Numerals	០	១	២	៣	៤	៥	៦	៧	៨	៩	Punctuations	។	៕	៖	ៗ	៘	៙	៚	៛	៝	៞	៟	០	១	◌ៈ
Codepoint	C9	CA	CB	CC	CE	CF	D0	CD	CD	Codepoint	E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	Codepoint	D9	DA	D4	D5	D8	D6	DB	D7	DC	DD					

(a) The Unicode point representation of the entire Khmer alphabet, including consonants, vowels, diacritics, and punctuation. Adapted from [1].

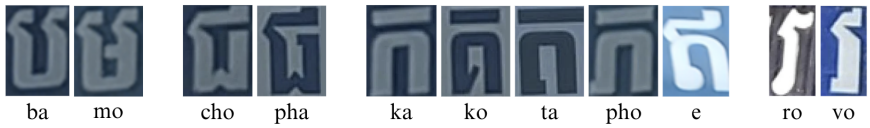
ស្តេច "King"	ស	◌្ត	◌េ	ច	ស្តាយ "Regret"	ស	◌្ត	◌ា	យ	មេត្តា "Mercy"	ម	◌េ	ត	◌្ត	◌ា	វត្តិ "Pagoda"	វ	ត	◌្ត
Codepoint	9F	D2	8A	C1	85	Codepoint	9F	D2	8A	B6	99	Codepoint	98	C1	8F	D2	8F	B6	Codepoint	9C	8F	D2	8F

(b) The Unicode point representation of word forms using "coeng (U17D2)". For the first two words, they use the subscript DA; for the last two words, they use the subscript TA, but both appear the same, while the Unicode points are different.

Fig. 1: Illustration of the complexity in how Unicode points encode the Khmer alphabet and words.

characters). Existing STDR methods, often designed for Latin scripts, struggle with non-Latin scripts like Khmer, which feature complex characteristics such as character stacking, diacritics, ligatures, non-uniform character widths, and the absence of explicit word boundaries. These complexities, combined with the low-resource nature of many non-Latin languages, necessitate advanced solutions.

In regard of non-Latin scripts like Khmer, the current state-of-the-art approaches in STDR struggle to fully address the aforementioned challenges. One of the main reasons is the complexity of Khmer characters, which consist of 33 consonants, 16 dependent vowels, and 14 independent vowels and 13 diacritics [7]. Khmer symbols are encoded using Unicode (U1780-U17FF), but the order of codes does not always follow the left-to-right writing direction. The relationship between symbols and codes is complex, with some symbols represented by multiple codes and some codes representing combinations of symbols. Since the subscripts don't have their own code, Khmer words are formed using a sequence of two codes: the "coeng (U17D2)" code, which is combined with the main consonant. Distinct from other languages, recognizing a Khmer word requires considering the entire word's writing, emphasizing on the importance of spatial information for accurate recognition. As specified by the Guinness World Records, Khmer script has the longest alphabet, consisting of 74 distinct characters [23]. Fig. 1 shows about the complexity of how Unicode point representation of Khmer alphabets and words. Depending on the fonts used, certain pairs of characters can be highly ambiguous, differing only by a single stroke. In extreme cases, some subscript forms are almost indistinguishable. Additionally, while some characters are formed as a single continuous glyph, others comprise multiple disconnected glyphs, each representing a distinct character. Furthermore,



(a) Different characters having similar appearances

(b) Same characters having different appearances

Fig. 2: The characters in (a) and (b) show the low inter-class variance versus high intra-class, which highlights the difficulty in Khmer writing. Group (a) demonstrates different characters appearing in similar shapes, while group (b) shows characters that appear in different shapes, although having the same characteristics.

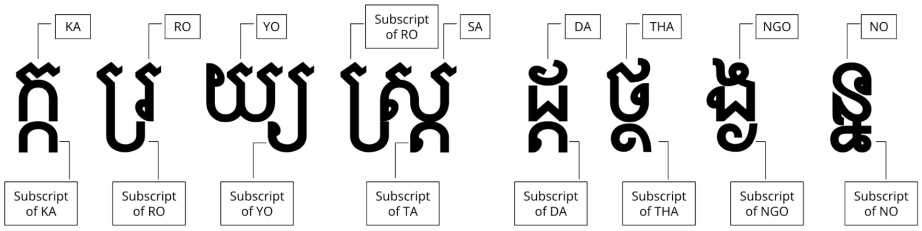


Fig. 3: Examples of multiple consonant clusters in Khmer, showing the different shapes of the consonants when they function as main or sub-consonants. Some consonants have only one sub-consonant, while others have more to form a word.

certain groups of characters appear similar in shape but are distinct, while others appear different due to the writing style and the different font types, as demonstrated in Fig. 2. Unlike Latin scripts, Khmer forms words differently. On the one hand, Khmer script has a distinctive feature: consonants can have different shapes based on their spatial position related to the word, including main and sub-consonants, can feature the merging of two or more consonants into different shapes, known as low-consonants or subscripts, which are located beneath the main consonant as illustrated in Fig. 3. This combination helps to create the desired consonant sound [25]. Vowels, on the other hand, can appear everywhere *i.e.* before, after, above, and below consonants. In some special cases, vowels can combine with each other to create new vowels. Additionally, the absence of word separation in Khmer writing makes the detection task even more challenging compared to languages that have distinct word boundaries. Moreover, challenges related to low contrast against complex backgrounds, varying lighting conditions, and non-standard text orientations contribute to the difficulty in achieving higher accuracy and reliability. These factors make Khmer text in scene images more difficult to detect and to accurately recognize.

In this work, we present the first dataset of Khmer text in natural scene images, named KhmerST (**Khmer S**cene-**T**ext). The dataset contains 1,544 expert-annotated images, including 997 indoor and 547 outdoor scenes, making it a diverse and complex collection. The challenges within this dataset include planar text, raised text, poorly illuminated text, distant text, and partially occluded text. Each scene image is annotated with line-level associated text and polygonal bounding box coordinates. We also benchmark the dataset using several state-of-the-art approaches for text detection and recognition. By introducing KhmerST, we aim to provide a valuable resource for future research on detection and recognition tasks, as well as text segmentation and word spotting of Khmer in natural scene images. The novelty of the KhmerST dataset lies in its comprehensive approach to capturing the uniqueness of the Khmer script in diverse real-world scenarios. This focus supports the development of more effective and inclusive computer vision technologies, filling a significant gap in resources tailored to Southeast Asian scripts, especially Khmer. Unlike most existing datasets that primarily focus on Latin, Chinese, or Arabic scripts, KhmerST provides an essential resource for creating solutions finely tuned to the needs of the Cambodian population. These applications include digital archiving of documents, automated translation services, and enhanced accessibility features for technology applications in Khmer. By capturing a variety of text appearances and settings not present in synthetic datasets, KhmerST serves as a more comprehensive and challenging resource for OCR development, enabling robust model training to handle a wide range of practical situations.

Therefore, this work makes two key contributions. First, we introduce KhmerST, a novel low-resource benchmark specifically designed for Khmer scene-text. Second, we demonstrate that the performance of current state-of-the-art models on KhmerST significantly lags behind their performance on Latin script benchmarks. This discrepancy underscores the necessity for more holistic and efficient modeling approaches tailored to the complexities of the Khmer script.

The paper is structured as follows. Sec. 2 reviews the related work. Sec. 3 introduces the KhmerST dataset and its characteristics, the detailed process of collecting the data, the annotation, while Sec. 4 presents the tasks, the baselines and the evaluation metrics, followed with a discussion. Finally, Sec. 5 offers a conclusion.

2 Related work

2.1 Datasets of Text in Natural Scene Images

As STDR is a popular research domain in the computer vision community, numerous dataset benchmarks are available in various languages. These datasets are classified into two main categories: real-world text and synthetic text. For real-world text datasets [13, 15, 26], they offer multilingual text images captured in natural scenes that contain text from urban environments and serve as a valuable benchmark for evaluating recognition algorithms. On the other hand, [3, 9] propose the synthetic text dataset that was generated using computer vision

techniques and produced a huge amount of labeled data. Besides, Google Street View images, which also count as a dataset for STDR belongs to the real-world text category. These datasets are particularly beneficial to facilitate the development of robust STDR systems. The Street View House Number (SVHN) dataset, presented by Netzer *et al.* [16], contains images of house numbers collected from Google Street View, providing a challenge for digit recognition due to the various fonts, sizes, and backgrounds of the images. The FSNS dataset, introduced by Smith *et al.* [21], consists of more than a million images from Google Street View in France. These images contain street name signs, which help address street name extraction problems. The CTW dataset, presented by Yuan *et al.* [28], contains a large collection of Chinese text with over 30,000 street view images. This dataset is an important resource for evaluating and developing scene text recognition systems for Chinese. The KAIST dataset, introduced by Jung *et al.* [10], contains Korean text information and image ground truth, encompassing a wide range of scene images that present text in different formats and contexts. For the Khmer language, there is a synthetic dataset proposed by Bouy *et al.* [1]. The dataset contains a large number of images of Khmer text. However, a comprehensive Khmer scene text dataset does not exist. The absence of such a dataset presents significant challenges for developing and evaluating scene text recognition systems for the Khmer language. Furthermore, we should have real-world data to better evaluate the capacity of existing systems on the Khmer script. Developing this dataset would fill a critical gap and advance optical character recognition (OCR) technology for future applications in Khmer.

2.2 Scene Text Detection and Recognition

The traditional approaches in scene-text detection focused on hand-crafted low-level features to differentiate text and non-text components in scene images. The sliding window (SW) method detects text information by moving the sub-window through all locations of the image; it utilizes the pre-train classifier to ensure whether text exists within the sub-window, as described by He *et al.* [6]. Wang *et al.* [27] conducted a convolutional neural network (CNN) with the SW method to find the position of text on the images. The connected component-based methods, as described by Zhu *et al.* [30], are designed to extract the components from the image and filter out non-text candidates using manually designed rules or automatically trained classifiers. Recently, there has been much attention on deep learning in semantic segmentation and general object detection. As a result, similar techniques are increasingly being utilized in text detection. Qin and Manduchi [17] developed a text detection method using a cascade of two convolutional neural networks (CNNs). Firstly, text regions of interest are identified by a fully convolutional network (FCN) and resized to a square shape with a fixed size. Redmon *et al.* [18] proposed YOLO (You Only Look Once), which is an object detection model that can predict the object and its location at a glance. It considers the entire image during training and test time, implicitly encoding contextual information about classes and their appearance.

In our case, for object detection, we propose to use different versions of YOLO and consider the text as an object.

As for the scene text recognition task, many approaches were considered in the literature such as character classification-based methods, word classification-based methods, sequence-based methods, end-to-end text spotting, *etc*. Lee and Osindero [11] proposed recursive recurrent neural networks (RNNs) enhanced with an attention model for text recognition. Jaderberg *et al*. [9] experimented with a CNN framework to train the synthetic data with handcrafted labeling and receive a good performance for word recognition. Li *et al*. [12] introduced transformer-based optical character recognition (TrOCR), an end-to-end text recognition approach with an image transformer and text transformer pre-train model. The TrOCR model can be used with large-scale synthetic data, printed text, and handwritten and scene text.

3 The KhmerST Dataset

The KhmerST dataset is a new collection specifically designed to advance computer vision research focused on the Khmer script. This dataset comprises numerous images captured from various public places in Cambodia, such as streets, signboards, supermarkets, and commercial establishments, all featuring text written in Khmer. To our knowledge, it is the first scene-text dataset for the Khmer language, making it a unique contribution compared to existing benchmark datasets that include Khmer printed text, as in [2,22], the historical handwritten Sleuk-Rith dataset [24], scanned books [4], synthetic documents, synthetic scene text, KHOB, and ID cards proposed by [1].The KhmerST dataset is crucial because it provides real-world scenarios, illustrating how the language is used in everyday contexts, essential for developing robust and accurate text recognition models. It addresses challenges such as varying lighting conditions, diverse font styles, and background noise typical in natural scene images. Unlike other datasets, the KhmerST dataset captures real-world environments, which is essential for training models to handle a wide range of practical situations. Additionally, it includes a variety of text appearances and settings not present in synthetic datasets, making it a more comprehensive and challenging resource for OCR development.

The novelty of the KhmerST dataset lies in its comprehensive approach to capturing the uniqueness of the Khmer script in diverse real-world scenarios. This focus supports the development of more effective and inclusive computer vision technologies, filling a significant gap in resources tailored to Southeast Asian scripts, especially Khmer. Unlike most existing datasets that primarily focus on Latin, Chinese, or Arabic scripts, KhmerST provides an essential resource for creating solutions finely tuned to the needs of the Cambodian population. These applications include digital archiving of documents, automated translation services, and enhanced accessibility features for technology applications in Khmer. By capturing a variety of text appearances and settings not present in synthetic datasets, KhmerST serves as a more comprehensive and challenging

Fig. 4: Examples of the KhmerST dataset: The Khmer script is present in both indoor and outdoor images. The dataset showcases the diversity of font sizes, styles, and the various ways the script appears, such as straight, rotated and curved text.

resource for OCR development, enabling robust model training to handle a wide range of practical situations.

3.1 Image Selection

To create the KhmerST dataset, we embarked on a data collection process across Cambodia, amassing a total of 1,544 images. These images were captured from a variety of locations to ensure a broad representation of settings. We utilized four different smartphone models for this purpose: Samsung Galaxy A32, iPhone 8 Plus, iPhone 13 Pro Max, and iPhone 14 Pro Max. This diversity in devices helped to capture images under different lighting conditions and camera capabilities, enhancing the dataset's robustness.

The KhmerST dataset is divided into two main categories: indoor and outdoor images. Indoor images feature text from commercial environments like supermarkets, while outdoor images include text from streets, signboards, and public buildings. Fig. 4 illustrates examples from both categories. The variety in font styles and the different ways the script appears (*e.g.* straight, rotated, and curved text) ensures that the dataset can be used to develop robust text detection and recognition models capable of handling various real-life conditions.

3.2 Dataset Annotation

The VGG Image Annotator (VIA) was utilized for the annotation process. It is a powerful tool designed for marking up images with annotations. This annotator allows to define regions within each image using polygon coordinates, effectively delineating complex shapes by specifying vertices on the x and y axes. These annotations are crucial for precise object detection and region-specific analysis.

Fig. 5: Examples of bounding boxes on text areas. The green bounding boxes represent polygon coordinates, while the yellow ones represent rectangular coordinates.

The data for each image, including its annotations, is structured in JSON format, offering a clear, hierarchical representation of attributes. In our JSON structure, each polygon's coordinates are represented by arrays of x and y points such as "all_points_x": $[x1, x2, x3, x4]$ and "all_points_y": $[y1, y2, y3, y4]$. This dataset format uses polygons linked to line-level text to describe text areas in images rather than rectangular coordinates, because polygons can adjust to the way the text appears (*e.g.* rotated text). This method accommodates text rotations and contours, improving recognition accuracy. The JSON entries also include essential metadata, such as image filenames and sizes, to enhance the dataset's utility for training and evaluating machine learning models, particularly for Khmer script recognition. Fig. 5 demonstrates the advantage of using polygons over rectangles for capturing text areas.

3.3 Dataset Splits

We divided our KhmerST dataset into training and test sets for the text detection task by allocating 80% of the 1,544 images to the training set and 20% to the test set. This split results in 1,236 training images and 308 testing images, combining both outdoor and indoor categories to enhance diversity and challenge. The KhmerST dataset is paired with images and JSON files containing detailed text information. For the text recognition task, we cropped the text regions from all images, yielding a total of 3,463 cropped images. We applied the same 80/20 split, resulting in 2,851 training images and 712 test images. This systematic division ensures a robust evaluation framework for both detection and recognition tasks.

4 Benchmark Tasks and Evaluation Metrics

In this section, we outline the performance of the baseline models along with the evaluation metrics on the proposed KhmerST dataset. The experiments were conducted on an NVIDIA RTX A6000 GPU and 252 GB of RAM. The process includes two modules: Text Detection and Text Recognition.

4.1 Scene Text-Line Detection

Given an input image, our challenge is to detect the bounding box regions of Khmer text at line-level. The ground truth for each text line's bounding box region is provided. To address this challenge, we experimented with four different YOLO models. The decision to utilize YOLO models for text-line detection is based on their unique strengths, which align well with the complexities of text recognition in varied and dynamic environments.

Evaluation Metrics. For the detection task, the Intersection over Union (IOU) metric is used to measure how well the predicted bounding boxes overlap with the actual ones. An IOU score ranges from 0 to 1, where 1 indicates a perfect match. Higher IOU score reflect more accurate object detection, while a lower score suggest inaccuracies in the predictions. To determine the count of one-to-one matches, we only consider region pairs with an IOU score above a defined threshold of 0.5. Consequently, we calculate the Detection Rate (DR), the Recognition Accuracy (RA), and the F-measure (FM) determined by combining DR and RA, following Eq. (1).

$$\mathcal{FM} = \frac{2.\mathcal{DR}.\mathcal{RA}}{\mathcal{DR}+\mathcal{RA}}; \quad \text{with} \quad \mathcal{DR} = \frac{o2o}{N}; \quad \text{and} \quad \mathcal{RA} = \frac{o2o}{M} \tag{1}$$

where, $o2o$ is the number of counting one-to-one matched pairs, N is the number of boxes in the ground truth, M is the number of bounding boxes detected by each baseline model. For the performance of pre-trained YOLO models YOLOv5, YOLOv8, YOLOv$_{10}$ are evaluated using common metrics such as precision, recall, mAP50, and mAP50-95. mAP50 measures the overlap between predicted and actual bounding boxes, considering a match correct if the overlap 50% or more. Meanwhile, mAP50-95 provides a more detailed accuracy assessment across varying overlap thresholds from 50% to 95%.

Baseline Models. First, the enhanced YOLOv$_1$ architecture is a single neural network designed to predict object class probabilities and bounding boxes simultaneously. It processes input images of 1472x1472 pixels with 3 channels and outputs a grid containing class probabilities and bounding boxes for each cell. The model, inspired by [18], is built as a convolutional neural network (CNN) and evaluated using the KhmerST dataset. It applies a series of convolutional layers, followed by LeakyReLU activation and max-pooling for down-sampling, gradually increasing the number of filters. The input images have a fixed size of 1472x1472 pixels with 3 channels, and the model produces a tensor of predictions with a size of 9x23x23 as the final output. The 23x23 represents the number of output grids, and 9 represents 1 for the existence of the object and 8 values for the polygon coordinates of x and y.

We experimented with the model as a modular list with different combinations of layer numbers (1, 2, 3) and filter amounts (8, 16, 24) randomly. As shown in Tab. 1, the reported results indicate that 2 layers with 24 filters produce the best detection rate of 0.733, a recognition accuracy of 0.866, and an F-Measure score of 0.794 compared to other variations of the model. The results of text-line detection using the enhanced YOLOv$_1$ with a CNN architecture are displayed

Table 1: The performance of the enhanced YOLOv$_1$+CNN architecture.

Nb. Layers	Nb. Filters	DR	RA	FM
1	8	0.694	0.819	0.751
2	24	0.733	0.866	0.794
3	24	0.627	0.828	0.714

Table 2: Performance comparison on different versions of YOLO models.

Model	Precision	Recall	mAP50	mAP50-95	Runtime(H)	Params(M)
YOLOv$_5$	0.847	0.787	0.875	0.591	0.56	7
YOLOv$_8$	0.873	0.832	0.899	0.625	1.293	11
YOLOv$_{10}$	0.905	0.76	0.87	0.593	1.965	8

in Fig. 6a. We observe that the bounding boxes are correctly positioned over the text areas, though in some cases, the model didn't predict the correct text area.

Second, we fine-tuned the pre-trained YOLOv5, YOLOv8, and YOLOv$_{10}$ on the KhmerST dataset. Each image was resized to 640x640 pixels, and the corresponding JSON files were pre-processed into a text file format containing the image filename and coordinates, including the class ID, *x_center*, *y_center*, *width*, and *height*, where class ID "0" represents Khmer text as an object. As shown in Tab. 2, YOLOv10 achieved the highest precision value of 0.905, while YOLOv8 excelled in recall, mAP50, and mAP50-90, with values of 0.832, 0.899, and 0.865, respectively. However, YOLOv10 recorded the lowest recall score of 0.760, compared to YOLOv5, which had a recall rate of 0.787, and 0.832 of YOLOv8. Overall, while YOLOv5 offers efficiency with balanced performance, YOLOv8 and YOLOv10 provide progressively more complex architectures and higher precision, albeit with increased computational demands and runtime. The Fine-tuned YOLO models predict the bounding region of each image by providing the bounding box values along with the mAP score. The results of detection using the fine-tuned versions of YOLOv$_5$ are shown in Fig. 6b, YOLOv$_8$ in Fig. 6c, and YOLOv$_{10}$ in Fig. 6d.

Overall, we observed that the best model among the YOLO versions is YOLOv8. This model achieved the highest recall, around 0.832, and a mean average precision (mAP) of 0.899, although its precision of 0.873 was slightly lower than that of YOLOv10, which had a precision of 0.905. However, YOLOv8 outperforms YOLOv10 in text detection due to several key factors. Firstly, YOLOv8 excels at detecting small objects [8], which is critical for text detection, as text often appears as small elements within images. Additionally, YOLOv8's 11M parameters allow it to capture more detailed features and model complexities, which are essential for accurately recognizing text with various fonts, sizes, and orientations. While YOLOv5, with its 7M parameters, and YOLOv10, with 8M parameters, may offer reduced latency, which comes at the cost of accuracy in text detection. In contrast, YOLOv8's ability to handle the intricacies of small

object detection makes it more suitable for this task, even with slightly higher latency, while it is also more efficient in terms of computational resources and time. Moreover, YOLOv1 also demonstrated a relatively good ability to detect text areas in images, with some limitations when dealing with images of low resolution and complex backgrounds, as illustrated in Fig. 6a.

(a) Detection results from YOLOv$_1$ enhanced with CNN: the model predicts the text area in each image by drawing bounding boxes.

(b) In the detection output of the YOLOv$_5$ model, we detect the text areas by drawing the bounding boxes with mAP values.

(c) In the detection output of the YOLOv$_8$ model, we detect the text areas by drawing the bounding boxes with mAP values.

(d) In the detection output of the YOLOv$_{10}$ model, we detect the text areas by drawing the bounding boxes with mAP values.

Fig. 6: The output of the YOLO models: For each image, YOLOv$_1$ predicts the text areas, and pre-trained models like YOLOv$_5$, YOLOv$_8$, and YOLOv$_{10}$ also produce mAP values. The images with green bounding boxes represent the ground truth, while those with yellow bounding boxes show the predictions from the model.

Table 3: Character Error Rate (CER) and Word Error Rate (WER) performances for TrOCR and Tesseract OCR.

Model	CER (%)	WER (%)
TrOCR	1.01	2.24
Tesseract	1.30	4.75

4.2 Scene Text Recognition

Scene Text recognition involves the identification and extraction of textual information. In this matter, we intend to investigate the performance of two different methodologies on our proposed KhmerST dataset.

Evaluation Metrics. For the recognition phase, we calculate the character error rate (CER) and word error rate (WER) as the ratio of unrecognized characters over the total number of characters in ground truth. It typically includes insertions, deletions, and substitutions. CER and WER is calculated as follows:

$$\mathrm{CER} = \frac{\mathrm{S}_c + \mathrm{D}_c + \mathrm{I}_c}{\mathrm{N}_c}; \quad \mathrm{WER} = \frac{\mathrm{S}_w + \mathrm{D}_w + \mathrm{I}_w}{\mathrm{N}_w} \tag{2}$$

where S The number of substitutions D is the number of deletions I: The number of insertions N The total number of characters in ground truth.

Baseline Methods. First, we experimented with the TrOCR pre-trained model, which consists of a Transformer-based encoder and an auto-regressive text Transformer decoder to perform optical character recognition (OCR). The TrOCR model is designed to understand the context and structure of written language. It has shown its superior performance compared to the current state-of-the-art models in printed, handwritten, and scene text recognition tasks [12]. TrOCR achieved a relatively good performance with an overall CER of 0.90 and WER of 1.02 as shown in Tab. 3. We calculated the CER and WER using Eq. (2). Second, we utilize the Tesseract tool, using its OCR capabilities to extract text from natural scene images. With the KhmerST dataset, Tesseract is able to achieve a CER of 1.30 and WER of 4.75. The results denoted in Tab. 4 show the outputs produced by the TrOCR pre-trained model and Tesseract. We randomly selected five images from the test set to demonstrate that the models were able to extract the text correctly in some cases, but failed to do so in others.

Overall, we observed that state-of-the-art models and tools such as the TrOCR pre-trained model and Tesseract tool, did not perform well with our KhmerST dataset. Both TrOCR and Tesseract faced significant difficulties in extracting Khmer scene text accurately. These challenges can be attributed to the unique characteristics of the Khmer script, including its complex ligatures, varying baseline, and intricate diacritics. Additionally, the presence of diverse backgrounds and varying text orientations in natural scenes further the recognition difficulties. Our observations highlight the need for more specialized OCR models that

Table 4: Examples of text recognition using the TrOCR model and Tesseract compared with the ground truth. Errors in the predictions are highlighted in red.

Instance	Ground-Truth	Tesseract	TrOCR
	ប្រមុំសិនរដូវភ្លៀង	នាក	២ ប្រមុំសិនរដូវភ្លៀង _ 2។យ ២ ព គ្រ	មើលនៅកម្ពុជាតិចជាងគេ
	ជួយអ្នកបើកបរ	ជួយអ្នករបើកបរ	សូមប្រយ័ត្នអរគុណ
	ថាមពលជាមួយគ្រុឌ	វាមឯយជាមយក្រុង	ក្រូចលក់ទូរស័ព្ទផ្សារ
	ទទួលបានទំហំកម្ចីខ្ពស់បំផុត	ទទួលបានទំហំកម្ចីខ្ពស់បំផុត |	ម៉ាស៊ីនស្លាញ់ផ្សារទំនើប
	សម្រាប់អាជីវកម្ម	សម្រាប់អាជីវកម	ចំណាកសម្រាប់រថយន្ត
	បន្ទប់សិក្សា ២	បន្ទបសក្សា ២	រហូតដល់
	៩វិច្ឆិកា ១៩៥៣	វិច្ឆិកា ០៩៥៣,	ដើមឆេឡក
	ម៉ោង 12 ថ្ងៃត្រង់ ដល់ 8យប់	ម៉ោ3 !?ខែទ្រទំល់!យ	លក់សៀវភៅនិងកីឡារិល
	ទៅរិមីព្រះមុន្នីរ៉េត	ទៅរិមីព្រះមុន្នីរ៉េត	គណបក្សប្រជាជនកម្ពុជា
	ខាងមុខមានការដ្ឋាន	ខាវ។៦មានការដ្ឋាន់	ក្នុងច្រកនេះបុគ្គលិក

can handle the complexity of the Khmer script, including the fonts and their appearances, as well as the diversity of scene images.

4.3 Limitations

The pre-trained TrOCR model, with its approximately 159 M parameters, requires a substantial amount of data for effective training and testing. Given the limited size of the KhmerST dataset, training the TrOCR model led to over-fitting. Over-fitting occurs because the model memorizes the training data rather than learning to generalize from it, resulting in poor performance on unseen data. Additionally, the large model size contributes to increased computational demands, making it less feasible for environments with limited resources. Conversely, Tesseract OCR, an open-source optical character recognition engine, performs well with well-resourced languages such as English [14, 19], Hindi [20], and Chinese [29]. These languages benefit from extensive annotated datasets and well-developed linguistic models, enabling Tesseract OCR to accurately extract text. However, for the low-resource Khmer language, Tesseract OCR's performance is sub-optimal. As shown in Tab. 4, the outputs from both the TrOCR model and Tesseract OCR did not align with the ground truth text for Khmer.

Several factors contribute to the poor performance of these state-of-the-art models with Khmer text in natural scene images. Firstly, the Khmer script is inherently complex, with intricate characters, multiple diacritics, and unique word formation rules. Unlike Latin-based scripts, Khmer characters combine in

various ways, creating numerous glyphs that must be recognized, as shown in Fig. 1. In addition, Khmer text in natural scenes often appears in diverse font styles, sizes, and orientations. This variability significantly challenges OCR models, as they need to adapt to different typographic representations of the same characters. Another issue is the lack of standardization in Khmer text, which can vary widely in terms of spelling, formatting, and orthographic conventions. This lack of standardization increases the difficulty of creating robust models that can handle all possible variations. Environmental factors also play a role: text in natural scenes is subject to various distortions due to lighting conditions, shadows, reflections, and occlusions. These factors can obscure parts of the text, making it harder for OCR models to accurately detect and recognize the text.

5 Conclusion

This research work introduces the KhmerST Dataset, the first scene text dataset for the Khmer language, which contains around 1,544 images. The dataset is divided into two main categories: indoor 997 images and outdoor 547 images. We annotated the text in the scene images at the line-level and stored the coordinates as polygons. The dataset was collected by capturing real images with various fonts, text sizes, and backgrounds. These present a significant challenge for text detection and recognition systems. We believe that our dataset will be a great resource for improving OCR and advancing research in Khmer STDR. In addition to the dataset, we also produced a text detection and recognition benchmark and discussed the performance limitations of the current state-of-the-art models on the KhmerST dataset. To deal with such a challenging dataset, STDR require different models that can handle the unique characteristics of the Khmer script and the diverse conditions in which the text appears.

Addressing the challenges of Khmer text recognition in natural scene images requires a multifaceted approach. Future research will focus on: (i) expanding the dataset with more diverse images and text styles to provide a comprehensive resource for the research community. Additionally, generating synthetic data will augment the existing KhmerST dataset, overcoming its size limitations; (ii) developing specialized architectures tailored to the intricacies of the Khmer script aims to enhance detection and recognition accuracy by incorporating specific linguistic and typographic features; (iii) emphasizing multimodal approaches that integrate visual and textual data will further refine text recognition capabilities, particularly in disambiguating complex text scenarios. These areas for future contributions are pivotal for advancing Khmer STDR in natural images, ensuring that OCR models effectively manage the intricacies and variations of this low-resource language.

Acknowledgments. This research study is supported by the France Government Scholarship, co-funded by the Cambodia Academy of Digital Technology (CADT). We would like to thank La Rochelle University, Laboratoire Informatique Image Interaction (L3i), for their funding in annotating the dataset.

References

1. 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.
2. 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.
3. Teófilo E de Campos, Bodla Rakesh Babu, and Manik Varma. Character recognition in natural images. In *International conference on computer vision theory and applications*, volume 1, pages 273–280. SCITEPRESS, 2009.
4. Yasuhisa Fujii, Karel Driesen, Jonathan Baccash, Ash Hurst, and Ashok C Popat. Sequence-to-label script identification for multilingual ocr. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, volume 1, pages 161–168. IEEE, 2017.
5. Lluís Gómez and Dimosthenis Karatzas. Textproposals: a text-specific selective search algorithm for word spotting in the wild. *Pattern recognition*, 70:60–74, 2017.
6. Tong He, Weilin Huang, Yu Qiao, and Jian Yao. Text-attentional convolutional neural network for scene text detection. *IEEE transactions on image processing*, 25(6):2529–2541, 2016.
7. Joshua Horton, Makara Sok, Marc Durdin, and Rasmey Ty. Spoof-vulnerable rendering in khmer unicode implementations. In *Proceedings of the Sixth Asian Conference on Information Systems*, pages 177–180, 2017.
8. Muhammad Hussain. Yolov5, yolov8 and yolov10: The go-to detectors for real-time vision. *arXiv preprint arXiv:2407.02988*, 2024.
9. Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227*, 2014.
10. Jehyun Jung, SeongHun Lee, Min Su Cho, and Jin Hyung Kim. Touch tt: Scene text extractor using touchscreen interface. *ETRI Journal*, 33(1):78–88, 2011.
11. Chen-Yu Lee and Simon Osindero. Recursive recurrent nets with attention modeling for ocr in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2231–2239, 2016.
12. Minghao Li, Tengchao Lv, Jingye Chen, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, and Furu Wei. Trocr: Transformer-based optical character recognition with pre-trained models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13094–13102, 2023.
13. Simon M Lucas, Alex Panaretos, Luis Sosa, Anthony Tang, Shirley Wong, Robert Young, Kazuki Ashida, Hiroki Nagai, Masayuki Okamoto, Hiroaki Yamamoto, et al. Icdar 2003 robust reading competitions: entries, results, and future directions. *International Journal of Document Analysis and Recognition (IJDAR)*, 7:105–122, 2005.
14. Ashee Mahajan, Anand Nayyar, Rachna Jain, and Preeti Nagrath. Natural scenes' text detection and recognition using cnn and pytesseract. In *The Fifth International Conference on Safety and Security with IoT: SaSeIoT 2021*, pages 159–171. Springer, 2022.
15. Anand Mishra, Karteek Alahari, and CV Jawahar. Scene text recognition using higher order language priors. In *BMVC-British machine vision conference*. BMVA, 2012.

16. Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 7. Granada, Spain, 2011.
17. Siyang Qin and Roberto Manduchi. Cascaded segmentation-detection networks for word-level text spotting. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, volume 1, pages 1275–1282. IEEE, 2017.
18. Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.
19. Saurabh Saoji, A Eqbal, and B Vidyapeeth. Text recognition and detection from images using pytesseract. *J Interdiscip Cycle Res*, 13:1674–1679, 2021.
20. Kumar Shwait, Preetpal Kaur Buttar, and Rahul Gautam. Detection and recognition of hindi text from natural scenes and its transliteration to english. *International Journal of Advanced Research in Computer Science*, 13(2), 2022.
21. Raymond Smith, Chunhui Gu, Dar-Shyang Lee, Huiyi Hu, Ranjith Unnikrishnan, Julian Ibarz, Sacha Arnoud, and Sophia Lin. End-to-end interpretation of the french street name signs dataset. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part I 14*, pages 411–426. Springer, 2016.
22. 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.
23. Dona Valy, Michel Verleysen, and Sophea Chhun. Text recognition on khmer historical documents using glyph class map generation with encoder-decoder model. In *ICPRAM*, pages 749–756, 2019.
24. Dona Valy, Michel Verleysen, and Sophea Chhun. Data augmentation and text recognition on khmer historical manuscripts. In *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 73–78. IEEE, 2020.
25. 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.
26. Andreas Veit, Tomas Matera, Lukas Neumann, Jiri Matas, and Serge Belongie. Coco-text: Dataset and benchmark for text detection and recognition in natural images. *arXiv preprint arXiv:1601.07140*, 2016.
27. Tao Wang, David J Wu, Adam Coates, and Andrew Y Ng. End-to-end text recognition with convolutional neural networks. In *Proceedings of the 21st international conference on pattern recognition (ICPR2012)*, pages 3304–3308. IEEE, 2012.
28. Tai-Ling Yuan, Zhe Zhu, Kun Xu, Cheng-Jun Li, Tai-Jiang Mu, and Shi-Min Hu. A large chinese text dataset in the wild. *Journal of Computer Science and Technology*, 34:509–521, 2019.
29. Zhang Yun-An, Pan Ziheng, Dui Hongyan, and Bai Guanghan. Yolov3-tesseract model for improved intelligent form recognition. *Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science)*, 14(6):1833–1842, 2021.
30. Yingying Zhu, Cong Yao, and Xiang Bai. Scene text detection and recognition: Recent advances and future trends. *Frontiers of Computer Science*, 10:19–36, 2016.