Cross-Lingual Learning for Low-Resource Khmer Scene Text Detection and Recognition

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

Cross-Lingual Learning for Low-Resource Khmer Scene Text Detection and Recognition

Vannkinh Nom[1,2], Saly Keo[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), France
[2] Cambodia Academy of Digital Technology (CADT), Cambodia
{vannkinh.nom, keo.saly, souhail.bakkali, muhammad_muzzamil.luqman, mickael.coustaty, jean-marc.ogier}@univ-lr.fr

Abstract. Scene text detection and recognition in low-resource languages such as Khmer pose significant challenges due to the scarcity of annotated datasets and the script's inherent complexity, including stacked consonants, intricate ligatures, and context-dependent vowel diacritics. Unlike Latin-based scripts, Khmer lacks clear word boundaries and exhibits a wide variety of character combinations, making it difficult for standard OCR systems to perform reliably. While recent advances in deep learning have achieved strong performance for high-resource languages, their direct applicability to Khmer remains limited and underexplored. This study investigates the potential of cross-lingual transfer learning as a solution to bridge this gap. By leveraging pretrained models from high-resource languages such as English, we explore how existing state-of-the-art detection and recognition architectures can be adapted to the Khmer script. Specifically, we evaluate the effectiveness of fine-tuning these models, originally trained on large-scale Latin or multilingual datasets, using a limited amount of annotated Khmer data to boost accuracy and generalization.

Keywords: Cross-lingual learning (CLL) · Khmer Script · Scene Text Detection and Recognition (STDR)

1 Introduction

Scene Text Detection and Recognition (STDR) is the process of automatically identifying and interpreting textual content that appears within natural images. It typically involves two sequential tasks: text detection, which locates and localizes regions of interest containing text, and text recognition, which extracts the textual content within those regions. STDR has emerged as a critical computer vision task with widespread applications across domains such as autonomous driving, augmented reality, and assistive technologies for visually impaired individuals [21,44]. These systems have found practical use in numerous real-world scenarios, including document analysis [11,33], word spotting in historical documents [30], automatic license plate recognition [10,42], multilingual sign translation [6], and content-based image retrieval [39]. Despite the remarkable progress

ថ្ងៃព្រហស្បតិ៍

ថ + ្ង + ៃ + ព + ្រ + ហ + ស + ្ប + ត + ិ + ៍

Fig. 1: A sample of the complex Khmer word "ថ្ងៃព្រហស្បតិ៍" (Thursday) illustrates a combination of character types: consonants are highlighted in red, sub-consonants in blue, vowels in green, and diacritics in orange.

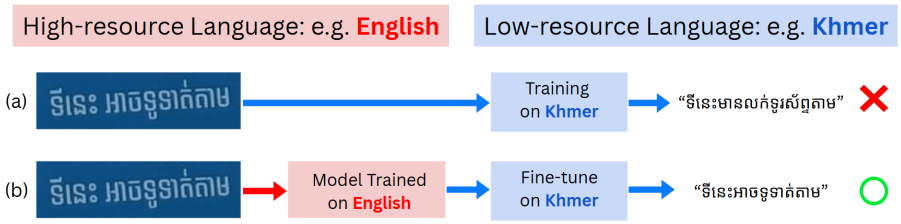


Fig. 2: Example of scene text recognition in Khmer, a low-resource language. (a) Training solely on Khmer data produces an incorrect result, while (b) applying CLL from a high-resource language like English enables correct recognition.

in STDR research, the vast majority of existing approaches have concentrated on high-resource languages such as English, Chinese, etc., which benefit from abundant training data and extensive research attention [1,18]. These methods typically leverage large-scale datasets such as ICDAR, MLT, and Total-Text to achieve impressive performance through sophisticated deep learning architectures [7,25]. However, this focus on high-resource languages has created a significant research gap for low-resource languages, where the scarcity of annotated data and unique linguistic characteristics present substantial challenges in developing effective STDR systems [22,23]. The performance disparity between high-resource and low-resource language STDR systems remains a critical barrier to achieving truly multilingual text understanding capabilities.

Khmer, the official language of Cambodia, exemplifies the challenges faced by low-resource languages in STDR applications. Although there are some existing datasets, including historical manuscripts [38], synthetic datasets [5], printed text corpora [37], and scene text datasets [27,28], the overall availability and diversity of annotated Khmer data remain limited. Khmer script poses unique challenges due to its complex writing system, which differs markedly from Latin. One key aspect is that consonants can change their forms based on their position within a word. This includes the use of sub-consonants that are written below, in front of, or behind the main consonant, and their shapes change depending on their position. Meanwhile, vowels in Khmer exhibit great positional flexibility, appearing before, after, above, or below consonants, and sometimes combining to create new vowel sounds. As shown in Fig. 1, Khmer words often feature clusters

of consonants, diacritics, and sub-consonants arranged in various positions relative to the base consonant, while vowels can occupy additional spaces above it. These complex combinations significantly increase the difficulty of accurately detecting and recognizing Khmer text. In addition, the Khmer script lacks explicit word boundaries, features variable character widths and heights, and includes numerous context-dependent character forms and ligatures [38]. These characteristics create substantial ambiguity in character segmentation and identification. Combined with the use of a wide variety of Khmer fonts and the frequent overlap of character components, these linguistic and visual properties make Khmer STDR considerably more difficult than systems designed for high-resource languages. Furthermore, the development of STDR systems for Khmer is hindered by the scarcity of annotated datasets. Unlike high-resource languages that benefit from large and diverse training corpora, Khmer suffers from data limitation. To address these challenges, we explore the potential of cross-lingual learning (CLL) as a promising solution for low-resource STDR. By leveraging pretrained models trained on large-scale datasets in high-resource languages such as English, we aim to adapt and fine-tune these models to the Khmer script using a limited amount of annotated data. CLL enables the reuse of visual and linguistic features learned in one language domain and facilitates knowledge transfer to another, potentially reducing the dependency on extensive labeled data while improving performance on underrepresented scripts. Fig. 2 illustrates this concept, showing how training solely on Khmer data can produce incorrect results, while applying CLL from a high-resource language like English enables correct recognition of Khmer scene text.

In this study, we investigate the effectiveness of fine-tuning state-of-the-art detection and recognition models compared to training from scratch on the same dataset, with the goal of identifying the most efficient strategy for Khmer STDR in low-resource settings. Specifically, our objective is to assess how the use of pre-trained models, originally developed for high-resource languages, impacts performance in terms of accuracy, generalization when applied to the Khmer script. Our evaluation includes both detection and recognition tasks to determine whether cross-lingual transfer learning offers a practical and scalable solution to improve STDR systems in underrepresented languages.

In this work, we present two key contributions. First, we introduce cross-lingual transfer learning for Khmer STDR, demonstrating how pretrained models developed for high-resource languages can be effectively adapted to low-resource like Khmer. Second, we conduct a comparative analysis of fine-tuning versus training from scratch, highlighting the practical benefits of leveraging pretrained knowledge in low-resource STDR scenarios.

2 Related Work

2.1 Text Detection and Recognition in Scene Images

Recent advancements in scene text detection and recognition have been largely driven by deep learning, particularly convolutional neural networks (CNNs) and

sequence models. For text detection, models such as EAST [47] and CRAFT [2] enable robust localization by directly regressing bounding boxes or character affinity fields, while DBNet [19] introduces differentiable binarization to improve detection accuracy in densely packed or curved text scenarios. Object detection frameworks like YOLO have also been adapted for text detection due to their efficiency in single-pass inference. For instance, YOLOv2 has been used to detect Chinese text [46], and recent studies [27,28] have explored various YOLO versions for Khmer text detection in natural scenes. For text recognition, early models such as CRNN [35] combine CNNs with bidirectional LSTMs and CTC loss to handle unsegmented sequences. More recent approaches employ attention mechanisms and Transformer-based architectures (e.g., TRBA, SRN [41]), which better capture long-range dependencies and support recognition in complex layouts. However, most of these methods have been developed and evaluated primarily on Latin-script datasets, limiting their generalization to structurally diverse scripts. In the context of Khmer, several studies have evaluated different recognition models. Author [27] explored Tesseract OCR and TrOCR for Khmer text line recognition. Author [28] investigated CRNN, VGG-based models, ResNet-based models, and Transformer-based architectures. Author [5] evaluated baselines such as CRNN, TRBA, TRBC, and also proposed a 2D recognition method tailored to the structural complexity of the Khmer script.

2.2 Low-Resource Language for STDR

Despite the remarkable progress in STDR for high-resourced languages, low-resource languages present substantial obstacles that remain largely unaddressed. Many writing systems, particularly in Southeast Asia and South Asia, are characterized by intricate structural properties including stacked characters, complex ligatures, and minimal inter-word spacing, as exemplified by Khmer, Devanagari, Tamil, Lao, and Burmese scripts. The fundamental constraint limiting progress is the scarcity of comprehensively annotated datasets, which creates significant barriers to developing robust models through conventional training approaches [32,14].

Some efforts have explored synthetic data generation and semi-supervised learning techniques for these underrepresented scripts, though such initiatives remain constrained in coverage and effectiveness [15]. Several benchmark datasets, including IndOCR [3], MLT-2019 [26], and ICDAR competitions [23], have made initial attempts to incorporate Indian and Southeast Asian scripts into their evaluation frameworks. However, scripts like Khmer, Lao, Burmese, and Nepali continue to be notably underrepresented in these datasets. The RRC-MLT dataset includes some Southeast Asian scripts but lacks sufficient diversity in text styles and imaging conditions for robust model training [36,40]. Consequently, the computational performance of state-of-the-art models on these scripts is infrequently documented, creating substantial knowledge gaps in the literature. This limitation is particularly problematic for real-world applications where these languages are used in multilingual environments, requiring systems that can handle script mixing and language identification challenges [17,12].

2.3 Cross-Lingual Transfer Learning

Cross-lingual transfer learning has emerged as a promising approach for addressing the low-resource language challenge by leveraging knowledge from high-resource languages [31,9]. The fundamental premise is that linguistic and visual patterns learned from abundant data in high-resourced languages can be adapted to improve performance in target languages with limited training data [29,43]. Recent research has extended cross-lingual transfer learning to computer vision tasks, particularly for document analysis and OCR. Multi-script OCR systems have been developed that can leverage training data from multiple languages simultaneously [23,36]. The author [40] demonstrated that pre-training on large-scale synthetic data from high-resource languages followed by fine-tuning on limited real data from target languages can significantly improve recognition accuracy. Similarly, the author [17] showed that transfer learning from English to Indian languages could achieve substantial performance gains even with minimal target language data. The effectiveness of cross-lingual transfer learning depends on several factors, including the similarity between source and target languages, the quality and quantity of target language data, and the architectural choices for adaptation [12,34]. Recent work has explored various adaptation strategies, including feature-based transfer, fine-tuning approaches, and domain adaptation techniques [8,24].

2.4 Research Gap and Motivation

Although deep learning-based methods have significantly advanced STDR for high-resource languages, several critical gaps remain for underrepresented scripts like Khmer. Despite its unique structural properties, Khmer is largely absent from major STDR benchmarks, and existing datasets lack scale and diversity compared to their Latin-script counterparts. While cross-lingual transfer learning has shown potential in document OCR, its effectiveness for complex scripts such as Khmer remains underexplored, and systematic evaluation frameworks for low-resource scripts are lacking.

This research addresses these challenges through a CLL framework designed to transfer knowledge from high-resource languages to Khmer, mitigating scarcity of annotated data. For text detection, we adopt YOLOv11 due to its superior performance, robustness in handling diverse scene conditions, and lightweight design suitable for deployment on resource-constrained devices. Previous work has shown that among several YOLO variants, YOLOv11 consistently outperforms others in detecting Khmer scene text in complex layouts [28]. We evaluate the model using both training from scratch and fine-tuning from pretrained weights on COCO dataset to assess cross-lingual transfer effectiveness.

For the recognition task, we select TrOCR for its Transformer-based encoder-decoder architecture, which enables learning rich contextual representations from large-scale text corpora. TrOCR eliminates recurrent components, providing both high recognition accuracy and computational efficiency [20]. Based on previous work on Khmer OCR baselines, which include CRNN, TRBC, TRBA,

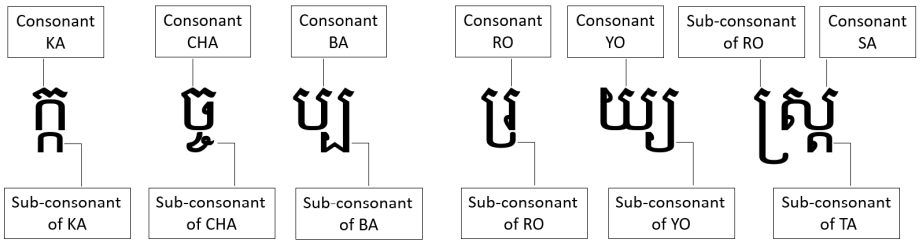


Fig. 3: Examples of multi-level Khmer consonant clusters, showing how consonants and sub-consonants change shape based on their function. Some clusters contain a single sub-consonant, while others combine multiple to form complex syllables or complete words.

and TrOCR, the TrOCR model demonstrated the best performance among the evaluated approaches. In this study, we experiment with TrOCR by systematically comparing training from scratch with fine-tuning on models pretrained on high-resource languages, in order to evaluate the effectiveness of CLL for Khmer scene text recognition.

3 Khmer Script and Dataset Overview

3.1 Khmer Script

Khmer serves as the official language of Cambodia and is spoken by around 17 million people. The language has been significantly influenced by Sanskrit and Pali due to Cambodia's historical ties to Hinduism and Buddhism [4]. The Khmer writing system follows an abugida structure, where consonants inherently carry an unwritten vowel sound [16]. The script consists of 33 consonants, 14 independent vowels, 23 dependent vowels, and 8 diacritics. These characters are encoded in the Unicode standard within the range U+1780 to U+17FF [13]. Khmer follows a left-to-right writing direction, with optional spaces added to enhance readability. The text consists of orthographic syllables formed by character clusters, each containing a primary consonant or a standalone vowel, potentially two subscript consonants, a vowel dependent on the base character, and diacritical marks [38]. In contrast to Latin writing, Khmer constructs words through a unique structural approach where consonants change their appearance based on their positional role within character clusters. Main and sub-consonants can merge to create altered forms known as subscripts or low-consonants, which appear beneath the base consonant as demonstrated in Fig. 3. This system creates dramatic size disparities within clusters, with subsidiary elements like subscripts, diacritics, and some vowels being markedly smaller than the foundational consonant [5]. Furthermore, font selection can create significant confusion between character pairs that vary by only one stroke mark. In severe instances, subscript variations become nearly impossible to differentiate. Additionally, character construction varies, with some formed as unified glyphs while others consist

```
"regions": [
  {
     "shape_attributes": {
        "name": "polygon",
        "all_points_x": [104, 827, 824, 97],
        "all_points_y": [538, 628, 749, 670]
     },
     "region_attributes": { "Name": "បាយស្រូបបឋម មីស៊ុបបឋម" }
  },
  {
     "shape_attributes": {
        "name": "polygon",
        "all_points_x": [1227, 1433, 1435, 1231],
        "all_points_y": [735, 733, 837, 843]
     },
     "region_attributes": { "Name": "អេស៊ីលីដា" }
  }
]
```


Fig. 4: An example of an annotated image paired with its corresponding JSON metadata. The metadata includes polygon coordinates for text regions and their transcriptions, supporting scene text detection and recognition tasks.

of separate, disconnected elements representing individual characters [27]. These inherent complexities in character formation, clustering, and visual similarity make Khmer script substantially more challenging for automated detection and recognition than Latin script's standardized character system.

3.2 Dataset

Since Khmer is a low-resource language, existing datasets still have limitations. Some public existing datasets focus mainly on printed text [37], synthetic datasets [5], and historical manuscripts [38]. Meanwhile, existing datasets for STDR include only KhmerST [27] and WildKhmerST [28]. Due to this limited availability of Khmer scene text datasets, we selected to use both datasets: WildKhmerST for training and KhmerST for evaluation.

The KhmerST dataset is the first publicly available dataset for Khmer scene text, comprising 1,544 annotated images, including 997 indoor and 547 outdoor scenes. It presents a wide range of real-world challenges, such as flat and raised text, low-light conditions, and text that is distant or partially obscured. Each image includes line-level annotations with corresponding transcriptions. The dataset also captures various text orientations, including horizontal, vertical, and curved layouts, reflecting typical use cases in storefronts, street signs, and advertisements.

The WildKhmerST dataset is a large-scale Khmer scene text dataset comprising 29,601 annotated text lines from 10,000 unique images captured in challenging real-world environments. This dataset addresses the shortage of training data for Khmer OCR applications, particularly for deep learning approaches. The collection features diverse scenarios, including artistic typography, blurred

text, low-light conditions, curved text arrangements, complex backgrounds, and partially occluded text elements. The dataset captures text as it appears in authentic public settings, such as street signs, shop banners, product labels, and advertisements, reflecting the variability and complexity of real-world usage.

Both datasets provide annotations that include polygon-based bounding box coordinates for text localization and ground truth transcriptions for recognition tasks, making them suitable for comprehensive scene text detection and recognition evaluation. The polygon annotations are particularly valuable for Khmer text, as they can accurately capture the irregular boundaries of curved or rotated text instances commonly found in real-world scenarios. This annotation detail enables precise localization training and supports development of more robust text detection algorithms that can handle geometric complexities inherent in scene text. Fig. 4 illustrates an example of image-text annotation, where the image is paired with a JSON file that specifies the coordinates of text regions using polygons along with their associated transcriptions in Khmer script.

4 Khmer STDR - Models and Performance

4.1 Scene Text Detection

Detecting text lines in Khmer script presents unique challenges due to the script's complex character structures, varying orientations, and the presence of cluttered or low-contrast backgrounds in natural scene images. To address these issues, we leverage the YOLO object detection framework, known for its real-time performance and adaptability to dense prediction tasks. Previous research has demonstrated the effectiveness of YOLO in similar contexts, with [45] proposing YOLOv2 for detecting Chinese characters in scene images, and [27] applying different YOLO variants to Khmer text-line detection. In this work, we investigate two distinct experiments using the YOLOv11 framework: (1) training a model from scratch with Khmer-specific initialization, and (2) fine-tuning a pretrained YOLOv11s[1] model originally developed for general object detection. For both experiments, we standardized input images to 800 * 800 pixels and represented text-line annotations using normalized coordinates (x_center, y_center, width, height) with class label '0' for Khmer text lines. Performance was evaluated using precision, recall, mAP50, and mAP50-95 metrics, with the latter providing comprehensive assessment across multiple IoU thresholds. All models were trained on the WildKhmerST dataset and evaluated on the KhmerST dataset.

In the first experiment, we implemented a from-scratch training approach using a custom YOLOv11 architecture configured specifically for Khmer text detection. The model maintained the standard YOLOv11 design with a lightweight backbone and detection head, optimized for efficient performance in resource-constrained scenarios. We configured the architecture for single-class detection by setting the number of classes parameter to 1 (labeled as "text"), while preserving all other structural components. This focused configuration enabled the

[1] https://docs.ultralytics.com/models/yolo11 Accessed: 2025-06-05

Table 1: Detection performance and efficiency comparison of YOLOv11 models trained from scratch (FS) and fine-tuned (FT) for Khmer text line detection. Metrics include precision, recall, mAP scores, FLOPs, and training time.

Model	Precision	Recall	mAP50	mAP50-90	FLOPs (G)	Training Hours (h)
YOLOv11 (FS)	0.807	0.801	0.852	0.565	113.7	63
YOLOv11 (FT)	**0.840**	**0.809**	**0.889**	**0.583**	**21.3**	**24**

model to dedicate its full capacity to learning Khmer text line patterns without the complexity of multi-class discrimination. The training process took approximately 63 hours and involved 113.7 GFLOPs per forward pass, reflecting the computational cost of learning from scratch. The model achieved 0.807 precision, 0.801 recall, 0.852 mAP50, and 0.565 mAP50-95, as summarized in Table 1.

In the second experiment, we employed transfer learning by fine-tuning a YOLOv11s model pretrained on the COCO dataset. We retained the model's robust low-level feature extraction capabilities from general object detection while adapting only the final layers for Khmer text line recognition. This strategy effectively leveraged the model's existing visual understanding while specializing it for our specific domain task. The fine-tuned model achieved higher detection performance while being significantly more efficient, requiring only 24 hours of training and 21.3 GFLOPs per forward pass. It is important to note that the reported 24-hour training time represents only the fine-tuning phase on our Khmer dataset and does not include the pretraining phase of YOLOv11s on the COCO dataset. Evaluation results demonstrated strong performance, achieving 0.809 recall, 0.889 mAP50, and 0.583 mAP50-95, as shown in Table 1. Fig. 5 provides a qualitative comparison of the detection results from the two experiments. Image (1) shows the output of the model trained from scratch, while image (2) displays the result of the fine-tuned model. In each image, green bounding boxes indicate ground truth annotations, and yellow boxes represent the model's predictions.

Overall, both experiments explored the effectiveness of training strategies for Khmer text detection using the YOLOv11 architecture. Comparing from-scratch training with transfer learning highlights the trade-offs between domain-specific adaptation and leveraging pretrained knowledge. Training from scratch allowed the model to fully adapt to the visual characteristics of Khmer text but required significantly higher computational resources: over double the training time (63 vs. 24 hours) and more than five times the FLOPs per inference (113.7G vs. 21.3G). In contrast, the fine-tuned model achieved better performance across metrics, including higher precision, recall, and mAP, while demanding less training effort. The models demonstrated strong detection capabilities, with notable gains in precision and mAP when incorporating pretrained weights. These results underscore the advantage of initializing from general object detection models, even for specialized domains like Khmer text. The findings suggest that fine-tuning offers a more efficient, effective approach for scene text detection in low-resource languages, particularly when computational constraints are a concern.

Fig. 5: Visualization of detection results using YOLOv11. Image (1) shows the result from a model trained from scratch, while image (2) shows the result from a fine-tuned model. In each image, green boxes represent ground truth, and yellow boxes represent the model's predictions.

4.2 Scene Text Recognition

Scene text recognition focuses on interpreting textual content within images. Given a cropped text line region from ground truth data, the task aims to accurately transcribe the contained text. To achieve this, advanced models like TrOCR leverage a Transformer-based encoder paired with an auto-regressive text decoder, enabling robust Optical Character Recognition (OCR). TrOCR excels in understanding linguistic context and structural patterns, outperforming existing state-of-the-art methods across diverse text types, including printed, handwritten, and scene text [20]. In this work, we explore two different approaches using TrOCR: (1) training a model from scratch using a Khmer dataset, and (2) fine-tuning a TrOCR model originally trained on Latin-script languages and adapting it to Khmer. Our goal is to demonstrate the effectiveness of cross-lingual transfer learning, showing that a model trained on high-resource languages can be successfully adapted to low-resource languages like Khmer. We also compare the performance of the fine-tuned model against the one trained from scratch to assess the benefits and limitations of transfer learning in low-resource scene text recognition. Due to the challenges posed by the Khmer script, such as ambiguous consonants, subscripts, vowels, ligatures, character clusters, and other unique script features, we choose to evaluate model performance using

KHCWER[2]. This metric is specifically designed to handle character clusters by accounting for vowel combinations and script-specific corrections. In contrast, conventional metrics like Character Error Rate (CER) and Word Error Rate (WER), as implemented in tools such as Jiwer[3] and TorchMetrics[4], are unable to capture these nuances and may therefore misrepresent recognition performance for Khmer text. For example, consider the sentence "កុកហើលើមេឃ". Given that both the ground-truth segmentation and instance segmentations are identical ['កុក', 'ហើលើ', 'មេឃ'], one would expect perfect accuracy. However, Jiwer reports a CER of 0.17 and a WER of 1, while TorchMetrics yields a CER of 0.1 and a WER of 1. In contrast, KHCWER correctly interprets the character clusters and adjusts both CER and WER to 0, indicating no error.

$$\text{CER} = \frac{S + I + D}{N}, \quad \text{WER} = \frac{\text{Incorrect words}}{\text{Total words in ground truth}} \tag{1}$$

For all experiments, we used the WildKhmerST dataset for training and the KhmerST dataset for evaluation. In the first experiment, we trained TrOCR models from scratch specifically for the Khmer script. We constructed custom VisionEncoderDecoder models with a ViT-base encoder (google/vit-base-patch16-224-in21k) and a BART-style decoder configured from scratch with parameters tailored for the Khmer script. We evaluated two different tokenization approaches: the NLLB tokenizer (facebook/nllb-200-distilled-600M) and the XMLRoberta tokenizer. Both configurations were combined with a ViT image processor to construct custom TrOCRProcessors. This setup was designed to better support Khmer's complex script structure, including subscripts, ligatures, and vowel clusters. The training and evaluation annotations were parsed from plain text files and converted into structured datasets via a custom CustomOCRDataset class. Models were trained using the Seq2SeqTrainer API from Hugging Face Transformers, with the AdamW optimizer, a batch size of 16, and 100 training epochs. To ensure reproducibility, all random seeds were fixed. As shown in Table 2, the from-scratch models achieved CER and WER of 0.40 and 0.55 with the NLLB tokenizer, and 0.44 and 0.59 with the XMLRoberta tokenizer.

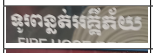
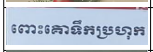
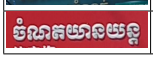
In the second experiment, we fine-tuned a pretrained TrOCR model (microsoft /trocr-small-printed) to assess the effectiveness of cross-lingual transfer learning, adapting a model originally trained on high-resource Latin script languages to low-resource Khmer script. We retained the original VisionEncoderDecoder architecture of the pretrained TrOCR model, preserving the pretrained ViT encoder and transformer decoder. Similarly to the first experiment, we evaluated two tokenization approaches: the NLLB tokenizer and XMLRoberta tokenizer, each combined with a ViT image processor to construct new TrOCRProcessors. For training, we employed the Seq2SeqTrainer API with the AdamW optimizer, a batch size of 16, and ran training for 100 epochs, matching the configuration from the first experiment. The results in Table 2 show that the

[2] https://github.com/keosaly/KHCWER.git Accessed: 2025-06-13
[3] https://pypi.org/project/jiwer Accessed: 2025-06-15
[4] https://lightning.ai/docs/torchmetrics/stable Accessed: 2025-06-15

Table 2: Performance comparison of TrOCR models trained from scratch (FS) versus fine-tuned (FT) with different tokenizers for Khmer text line recognition, with CER and WER calculated using Equation 1.

Model	Encoder	Decoder	Tokenizer	CER	WER
TrOCR (FS)	ViT-base-patch16-224-in21k	BART-style	NLLB	0.40	0.55
TrOCR (FT)	TrOCR-small-printed	Tr.Decoder	NLLB	**0.18**	**0.35**
TrOCR (FS)	ViT-base-patch16-224-in21k	BART-style	XMLRoberta	0.44	0.59
TrOCR (FT)	TrOCR-small-printed	Tr.Decoder	XMLRoberta	**0.17**	**0.33**

Table 3: Results of Khmer text recognition. Comparison of TrOCR (FS) trained from scratch and TrOCR (FT) fine-tuned models using two different tokenizers: NLLB and XMLRoberta, evaluated against the ground truth. Text highlighted in red indicates incorrect predictions made by the models.

Instance	Ground-Truth	TrOCR (FS) NLLB	TrOCR (FT) NLLB	TrOCR (FS) XMLRoberta	TrOCR (FT) XMLRoberta
	ម.វិថី សហព័ន្ធរុស្ស៊ី	ម.វិថីសហព័ន្ធរុស្ស	ម.វិថីសហព័ន្ធរុស្ស៊ី	ម.វិថីសហព័ន្ធរុស្ស	ម.វិថីសហព័ន្ធរុស្ស៊ី
	ទីនេះអាចទូទាត់តាម	ទីនេះអាចទូទាត់តាម	ទីនេះមានលក់ទូរស័ព្ទតាម	ទីនេះអាចទូទាត់តាម	ឌីអរទូទាំងប្រទេស	ទីនេះអាចទូទាត់តាម
	ទូរពន្លត់អគ្គីភ័យ	ទូរពន្លត់អគ្គីភ័យ	ទូរពន្លត់អគ្គីភ័យ	ទូរពន្លត់អគ្គីភ័យ	ទូរពន្លត់អគ្គីភ័យ	ទូរពន្លត់អគ្គីភ័យ
	ការិយាល័យ	ការិយាល័យ	ការិយាល័យ	ការិយាល័យ	ការិយាល័យ	ការិយាល័យ
	បានរង្វាន់ភ្លាមៗ	បានថ្នះឥឡូវនេះ!	បានរង្វាន់ភ្លាមៗ	បានស្វែងរក	បានរង្វាន់ភ្លាមៗ
	ពោះគោទឹកប្រហុក	អគារទឹកជ្រោនបុក	ពោះគោទឹកប្រហុក	អគារតាំងបង្ហាញ	ពោះគោទឹកប្រហុក
	សេវាកម្មធនាគារ	សេវាកម្មធនាគារ!	សេវាកម្មធនាគារ	សេវាកម្មធនាគារ	សេវាកម្មធនាគារ
	មានបន្ទប់សំរាប់ជូល	មានបន្ទប់សំរាប់ជូល	មានបន្ទប់សំរាប់ជូល	មានបន្ទប់សំរាប់ជូល	មានបន្ទប់សំរាប់ជូល
	បើកគណនីបញ្ជើ	បើកគណនីជ្រាក់បញ្ជើ	បើកគណនីបញ្ជើ	បើកគណនីថ្មី	បើកគណនីបញ្ជើ
	ចំណតយានយន្ត	ចំណតរថយន្ត	ចំណតយានយន្ត	ចំណតយានយន្ត	ចំណតយានយន្ត

fine-tuned models achieved significantly better performance: CER of 0.18 and WER of 0.35 with the NLLB tokenizer, and CER of 0.17 and WER of 0.33 with the XMLRoberta tokenizer. Table 3 provides visual examples that compare recognition performance of all four model configurations against ground truth. Fine-tuned models generally produce more accurate output, with incorrect predictions highlighted in red to clearly show where each approach fails. These qualitative results complement quantitative metrics by showing how different training strategies and tokenizers handle real Khmer text recognition tasks.

Overall, the comparative results in Table 2 demonstrate that fine-tuning pretrained models consistently outperforms training from scratch across both tokenization approaches, with improvements of 55-61% in CER and 36-44% in WER. The XMLRoberta tokenizer showed slightly better performance than

NLLB in both experimental settings, suggesting that tokenizer choice can influence recognition accuracy for Khmer script. Despite the script mismatch between the original Latin-script training data and the Khmer target script, transfer learning proved highly effective, underscoring the generalizability of learned visual-textual representations. These findings highlight the importance of leveraging pretrained models in low-resource scenarios and demonstrate the potential of cross-lingual OCR frameworks when appropriate domain-specific adjustments such as tokenization and processor customization are applied.

4.3 Discussion and Limitation

Scene Text Detection. The comparative results between training from scratch and fine-tuning reveal critical trade-offs in both computational efficiency and detection accuracy. Training the YOLOv11 model from scratch required a significantly higher computational budget, consuming 113.7 GFLOPs per inference and 63 hours of training. In contrast, fine-tuning a pretrained YOLOv11s model dramatically reduced both metrics, requiring only 21.3 GFLOPs and 24 hours of training. This represents a reduction of over 80% in FLOPs and more than 60% in training time. Despite the lower computational cost, the fine-tuned model achieved superior detection performance across all evaluation metrics, including higher mAP50 (0.889 vs. 0.852) and mAP50-95 (0.583 vs. 0.565), highlighting the effectiveness of leveraging pretrained features even for domain-specific tasks like Khmer text detection. This efficiency and performance trade-off underscores the practical value of transfer learning, especially in low-resource settings where computational constraints and limited labeled data are common. Fine-tuning not only accelerates model convergence but also benefits from generalized visual representations learned from large-scale datasets, which can be effectively repurposed for detecting text lines in complex scene contexts.

However, both models exhibited failure cases, particularly under challenging visual conditions. As shown in Fig. 6, common failure modes include missed detections of low-contrast text lines, horizontal text lines, and false positives in textured regions that visually resemble text. These issues are especially pronounced in scenes with cluttered backgrounds, stylized Khmer fonts, or extreme lighting variations.

While fine-tuning provides clear advantages in efficiency and overall performance, training from scratch may still be beneficial in scenarios where the target domain significantly diverges from the source domain (for example, highly stylized or script-specific text layouts). Ultimately, the choice between training strategies should be guided by a balance between performance goals, available computational resources, and the characteristics of the dataset.

Scene Text Recognition. Despite the strong performance gains achieved through cross-lingual fine-tuning, several limitations remain in the Khmer scene text recognition pipeline. First, although the pretrained TrOCR model was initially trained on high-resource Latin-script datasets, it transferred reasonably well to the Khmer domain. However, it still struggles to fully capture the complexity of

Fig. 6: Samples of detection failure cases. Images (1) and (3) display the ground truth bounding boxes highlighted in green, while the model predictions in images (2) and (4) show yellow boxes for correct detections and red for missed ones.

the Khmer script. The script includes features such as stacked consonants, overlapping vowel signs, and numerous diacritical marks, all of which pose significant challenges for encoder-decoder architectures. These structural complexities often lead to issues like merged characters or incorrect alignment during decoding, particularly in images with low contrast or visually cluttered backgrounds.

Another important limitation lies in the tokenizer design. Although both the NLLB and XMLRoberta tokenizers showed relatively strong performance, they are not specifically designed for the Khmer language. The small difference in performance between the two (with XMLRoberta slightly outperforming NLLB) indicates that a tokenizer tailored to the structure of Khmer could further enhance recognition accuracy. Current tokenizers treat Khmer text as a single block of characters, without accounting for important internal components such as consonant clusters or dependent vowels, which are essential for precise recognition.

From a training perspective, models trained from scratch underperformed significantly compared to fine-tuned ones. While this is expected in low-resource scenarios, it points to the difficulty of learning robust visual-textual mappings for complex scripts like Khmer without leveraging prior knowledge. However, both models showed notable failure cases, especially under challenging visual conditions. In particular, they struggled with freestyle handwriting and irregular horizontal text lines, which introduce high variability in character shapes, spacing, and alignment. This variability makes it difficult for the models to learn consistent visual-textual patterns. As illustrated in Fig. 7, freestyle writing often features uneven strokes, overlapping characters, and inconsistent baselines, while irregular horizontal lines can be skewed or distorted. These complexities frequently result in character misclassification or decoding errors. This limitation underscores the need for more diverse and representative training data, especially samples that include handwritten and stylized Khmer text. Future work could also explore the development of Khmer-specific tokenizers and hybrid architectures that better model the script's visual and linguistic structure.

Fig. 7: Sample of recognition failure cases. Challenges include freestyle handwriting with irregular shapes, horizontally aligned text with low contrast, and curved text lines, all of which contribute to reduced recognition accuracy.

Addressing these challenges is crucial for building more accurate and resilient OCR systems tailored to Khmer and other low-resource scripts.

5 Conclusion

In conclusion, this research work demonstrates the viability of cross-lingual transfer learning for addressing STDR challenges in Khmer, a low-resource language with complex linguistic characteristics. Our investigation reveals that pre-trained models from high-resource languages can be successfully adapted to handle the unique features of Khmer script, including its stacked consonants, intricate ligatures, and context-dependent vowel diacritics. The experimental results confirm that fine-tuning state-of-the-art architectures on limited annotated Khmer data yields significant improvements in both accuracy and generalization compared to training from scratch. This approach not only reduces data requirements but also leverages rich feature representations learned from extensive multilingual datasets. This is particularly valuable given the scarcity of large-scale annotated datasets for Khmer and similar low-resource languages. Our findings highlight both the promise and challenges of adapting existing OCR architectures to complex scripts with rich visual structures and limited linguistic resources. The work contributes to broader understanding of how deep learning models can be adapted across linguistically diverse scripts, moving beyond traditional focus on Latin-based languages and opening possibilities for developing OCR systems for other underrepresented scripts.

Future research will focus on developing model architectures specifically tailored to the structural and visual complexities of the Khmer script for STDR. This includes designing modules that more effectively handle character stacking, ligature composition, and diacritic placement. Additionally, we plan to enhance training by incorporating large-scale synthetic Khmer scene text datasets. This data augmentation strategy will improve model robustness and generalization to real-world conditions, particularly where annotated data is limited.

References

1. 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).
2. Baek, Y., Lee, B., Han, D., Yun, S., & Lee, H. (2019). Character region awareness for text detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 9365-9374).
3. A. Bhunia, S. Roy, A. Bhowmick, U. Pal, K. Roy, Y.-Z. Song, and U. Bhattacharya, "IndOCR: A New Benchmark for Indian Script OCR," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022.
4. Born, S., Valy, D., & Kong, P. (2022, December). Encoder-decoder language model for Khmer handwritten text recognition in historical documents. In 2022 14th International Conference on Software, Knowledge, Information Management and Applications (SKIMA) (pp. 234-238). IEEE.
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. Bušta, M., Patel, Y., & Matas, J. (2019). E2e-mlt-an unconstrained end-to-end method for multi-language scene text. In Computer Vision–ACCV 2018 Workshops: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers 14 (pp. 127-143). Springer International Publishing.
7. Ch'ng, C. K., & Chan, C. S. (2017, November). Total-text: A comprehensive dataset for scene text detection and recognition. In 2017 14th IAPR international conference on document analysis and recognition (ICDAR) (Vol. 1, pp. 935-942). IEEE.
8. Dabre, R., Chu, C., & Kunchukuttan, A. (2020). A survey of multilingual neural machine translation. ACM Computing Surveys (CSUR), 53(5), 1-38.
9. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019, June). Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers) (pp. 4171-4186).
10. Eltouny, K., Gomaa, M., & Liang, X. (2023). Unsupervised learning methods for data-driven vibration-based structural health monitoring: a review. Sensors, 23(6), 3290.
11. 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.
12. Ghiffari, F. A. A., Alfina, I., & Azizah, K. (2024). Cross-lingual transfer learning for Javanese dependency parsing. arXiv preprint arXiv:2401.12072.
13. Horton, J., Sok, M., Durdin, M., & Ty, R. (2017). Spoof-vulnerable rendering in Khmer Unicode implementations. In Proceedings of the Sixth Asian Conference on Information Systems (pp. 177-180).
14. Hedderich, M. A., Lange, L., Adel, H., Strötgen, J., & Klakow, D. (2020). A survey on recent approaches for natural language processing in low-resource scenarios. arXiv preprint arXiv:2010.12309.
15. Jatowt, A., Coustaty, M., Nguyen, N. V., & Doucet, A. (2019, June). Deep statistical analysis of OCR errors for effective post-OCR processing. In 2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL) (pp. 29-38). IEEE.

16. Kaing, H., Ding, C., Utiyama, M., Sumita, E., Sam, S., Seng, S., ... & Nakamura, S. (2021). Towards tokenization and part-of-speech tagging for Khmer: Data and discussion. Transactions on Asian and Low-Resource Language Information Processing, 20(6), 1-16.
17. Krishnan, P., Dutta, K., & Jawahar, C. V. (2018, April). Word spotting and recognition using deep embedding. In 2018 13th IAPR International Workshop on Document Analysis Systems (DAS) (pp. 1-6). IEEE.
18. Liao, M., Zhang, J., Wan, Z., Xie, F., Liang, J., Lyu, P., ... & Bai, X. (2019, July). Scene text recognition from two-dimensional perspective. In Proceedings of the AAAI conference on artificial intelligence (Vol. 33, No. 01, pp. 8714-8721).
19. Liao, M., Wan, Z., Yao, C., Chen, K., & Bai, X. (2020, April). Real-time scene text detection with differentiable binarization. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 07, pp. 11474-11481).
20. 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).
21. Long, S., He, X., & Yao, C. (2021). Scene text detection and recognition: The deep learning era. International Journal of Computer Vision, 129(1), 161-184.
22. Lyu, P., Yao, C., Wu, W., Yan, S., & Bai, X. (2018). Multi-oriented scene text detection via corner localization and region segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7553-7563).
23. Mathew, M., Karatzas, D., & Jawahar, C. V. (2021). Docvqa: A dataset for vqa on document images. In Proceedings of the IEEE/CVF winter conference on applications of computer vision (pp. 2200-2209).
24. Muller, B., Anastasopoulos, A., Sagot, B., & Seddah, D. (2020). When being unseen from mBERT is just the beginning: Handling new languages with multilingual language models. arXiv preprint arXiv:2010.12858.
25. Nayef, N., Yin, F., Bizid, I., Choi, H., Feng, Y., Karatzas, D., ... & Ogier, J. M. (2017, November). Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt. In 2017 14th IAPR international conference on document analysis and recognition (ICDAR) (Vol. 1, pp. 1454-1459). IEEE.
26. Nayef, N., Patel, Y., Busta, M., Chowdhury, P. N., Karatzas, D., Khlif, W., ... & Ogier, J. M. (2019, September). ICDAR2019 robust reading challenge on multi-lingual scene text detection and recognition—RRC-MLT-2019. In 2019 International conference on document analysis and recognition (ICDAR) (pp. 1582-1587). IEEE.
27. 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).
28. Nom, V., Keo, S., Bakkali, S., Luqman, M. M., Coustaty, M., Rossinyol, M., & Ogier, J.-M. (2025). WildKhmerST: A comprehensive dataset and benchmark for Khmer scene text detection and recognition in the wild. In Proceedings of the International Conference on Document Analysis and Recognition (ICDAR).
29. Pires, T., Schlinger, E., & Garrette, D. (2019). How multilingual is multilingual BERT?. arXiv preprint arXiv:1906.01502.
30. Rath, T. M., & Manmatha, R. (2007). Word spotting for historical documents. International Journal of Document Analysis and Recognition (IJDAR), 9, 139-152.
31. Ruder, S., Peters, M. E., Swayamdipta, S., & Wolf, T. (2019, June). Transfer learning in natural language processing. In Proceedings of the 2019 conference of the

North American chapter of the association for computational linguistics: Tutorials (pp. 15-18).
32. Ruder, S., Vulić, I., & Søgaard, A. (2019). A survey of cross-lingual word embedding models. Journal of Artificial Intelligence Research, 65, 569-631.
33. 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.
34. Schuster, S., Gupta, S., Shah, R., & Lewis, M. (2018). Cross-lingual transfer learning for multilingual task oriented dialog. arXiv preprint arXiv:1810.13327.
35. 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.
36. Singh, A., Pang, G., Toh, M., Huang, J., Galuba, W., & Hassner, T. (2021). Textocr: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 8802-8812).
37. 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.
38. 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.
39. Wang, K., Babenko, B., & Belongie, S. (2011, November). End-to-end scene text recognition. In 2011 International conference on computer vision (pp. 1457-1464). IEEE.
40. Wang, T., Zhu, Y., Jin, L., Luo, C., Chen, X., Wu, Y., ... & Cai, M. (2020, April). Decoupled attention network for text recognition. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 07, pp. 12216-12224).
41. F. Wang, H. Xie, Z. Zha, and S. Lu, "Scene Text Recognition with Transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
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, S., & Dredze, M. (2019). Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. arXiv preprint arXiv:1904.09077.
44. Ye, Q., & Doermann, D. (2014). Text detection and recognition in imagery: A survey. IEEE transactions on pattern analysis and machine intelligence, 37(7), 1480-1500.
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. 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.
47. 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).