



សាសនពិភាគអ៊យត្បូមិន្ទំពេញ

ROYAL UNIVERSITY OF PHNOM PENH

គារបំពេលរួមភាពនុវត្តកសាងទៅបានក្នុងក្រុមហ៊ុនខ្មែរនិងអង់គ្លេស ដោយប្រើប្រាស់បច្ចុប្បន្ន Craft បានក្នុងក្រុមហ៊ុន TrOCR

An End-to-End approach for Khmer & English Text  
Recognition using Craft with TrOCR Architecture

Mr. Vitou Soy

### A Thesis

In Partial Fulfilment of the Requirement for the Degree of  
Bachelor of Engineering in Information Technology Engineering

Examination committee: Mr. Sokchea Kor (Advisor)  
Mr. Champiseth Chap (committee)  
Mrs. Daly Chea (committee)  
Dr. ....

June 2025

## **SUPERVISOR's RESEARCH SUPERVISION STATEMENT**

Name of program: Bachelor of Engineering in Information Technology  
Engineering

Name of candidate: Vitou Soy

Title of thesis: An End-to-End approach for Khmer & English Text Recognition using Craft with TrOCR Architecture

This is to certify that the research carried out for the above titled master's research report was completed by the above named candidate under my direct supervision. This thesis material has not been used for any other degree. The candidate has demonstrated strong research capabilities and independence in developing novel approaches for Khmer text recognition. The research methodology, implementation, and results are original contributions to the field of Khmer OCR technology. I have provided guidance and oversight throughout the research process while allowing the candidate to explore innovative solutions.

Supervisor's name: Sokchea Kor

Supervisor's signature:.....

Date.....

ଶ୍ରୀମଦ୍ଭଗବତ

ក្នុងសហគមន៍បច្ចេកវិទ្យាតីមានសម្រួលដែលបានប្រើបានបច្ចុប្បន្ន យ៉ាងទូលំទូលាយ សម្រាប់បំលែង ឯកសារ សរសេរ ប្លូបការអក្សរខ្សោចាដអត្ថបទ នៃក្នុងបច្ចុប្បន្ន ( digital text ) ។ ការអក្សរខ្សោច ឬ OCR សម្រាប់ការសារឡើង ត្រូវបានប្រើបានបច្ចុប្បន្ន ដោយសារកង្ហ់ផែនប្រកបទទិន្នន័យ និងឯកសារសម្រាប់ train AI model ។ ដើម្បីធ្វើដោះស្រាយ បច្ចាន់នេះ យើងបានបង្កើតទិន្នន័យសិប្បនិមិត ( Synthetic Dataset ) ដោយប្រើបិទិន្នបច្ចេកទេសកម្រិតខ្ពស់ ។  
ក្នុងដំណឹងការបង្កើតទិន្នន័យសិប្បនិមិត ( Synthetic Dataset ) រួមមាន៖

- វិធីសាស្ត្រក្នុងការប្រមូលអត្ថបទចេញពីអ្ននដើរាជ មានជូចខាងក្រោម (Scrape data) :
    - ដំណាក់កាលទីមួយ៖ យើងបានប្រមូលអត្ថបទចេញពី khsearch.com, Chuon-Nath-Dictionary, Alpha-Word, Google-Word, និងបង្កើតក្រាយគឺ Huggingface.com ។
    - ដំណាក់កាលទីពី៖ យើងបានសម្ងាត់ ទិន្នន័យទាំងអស់នៅ៖ ផ្តើកតែតែដំណើរការ ដូចជា លុបពោលត្បូរអក្សរណាដែលមិនស្មើមាន ត្រូវមាននៅលើ រូបភាព ពីកញ្ចប់ និងបានលុបពោល ត្រូអក្សរណាដែល Fonts renders អត់ចេញ។
    - ដំណាក់កាលទីបី៖ ដំណាក់កាលមួយនេះ យើងបានធ្វើការ កាត់ប្រយោតទាំងអស់នៅ៖ ជាពាក្យ ដោយ របីប្រាស់ library ណូឡាំ៖ khmer-nltk
    - ដំណាក់កាលទីបូនេះ ចុងក្រាយ កំណានផ្សេបចំជាប្រយោតដែល មានប្រើប្រាស់ Random ពី ១ អក្សរ ហូតដល់ ១១០ អក្សរ ។
  - បង្កើតរូបភាពដោយអនុវត្តតាមលក្ខខណ្ឌខាងក្រោម :
    - ផ្តើខាងក្រាយចំណុច (Apply Different backgrounds)
    - បំពាក់ពុម្ពអក្សរផ្សេងៗគ្នា (Apply Different fonts)
    - Noise: gaussian\_noise, salt\_pepper\_noise, speckle\_noise, blur
    - បង្កើលអក្សរបន្ទិច (random rotation text)
    - បញ្ចូល Margin Randomly (1, 5) pixels
  - សរបចកយើងបានបង្កើត Data ជាង ៤ លាន records សម្រាប់ train OCR model

Architecture OCR គ្រប់ប្រព័ន្ធបង្ហាញ និង Text Recognition:

Text Detection: យើងប្រើមួន CRAFT ដោយបានធ្វើការ Train ឡើងវិញដោយ បាន annotation ទៅលើលើរូបភាពប្រាំហុល ៥០០ images និងសម្រាប់នឹង bounding box ជាង ៩០,០០០ boxes។

Text Recognition: យើងប្រើ TrOCR base model ចិត្តពី Microsoft (មាននៅក្នុង Hugging Face) ហើយបាន fine-tune ទៅលើ dataset ខ្លួនប្រព័ន្ធមូតិ (Synthetic Dataset) ដើម្បីបង្កើនសមត្ថភាពក្នុងការសម្ងាត់អគ្គរ៉ែខ្សោយ

លទ្ធផលសិក្សាបានបង្ហាញថា OCR របស់ពួកយើងអាចសម្ងាត់អត្ថបទចេញពីរបាយ បានដោយភាពត្រីមត្រី លើសពី ៤០%។ ជូនូវការនេះបង្ហាញថា ពីសក្សានុពលនៃការបង្កើត dataset នឹងការរឿបចំដែនាំដើម្បីអភិវឌ្ឍន៍ OCR ភាសាដំឡើងប្រសិទ្ធភាពកាន់តែខ្លស់។ កម្មាធិសមត្ថភាព អាចបានបំផុតអត្ថបទមិនត្រីមត្រី ពាក្យខ្លឹម ប៉ុណ្ណោះទេ តែវាកំអាចធ្វើការបានបំផុត ជូនូវជាមួយតួអក្សរដោយមួយតួអក្សរ, ពាក្យដោយពាក្យ, ប្រយោគដោយប្រយោគ ហូទុដល់ មួយប្រយោគដែល ១១០ តួអក្សរបែងចែកជាអ្នកដោយភាសាអង់គ្លេស។ ហើយលើសពីនេះទៀត វាកំអាចធ្វើការកំណត់សម្ងាត់ទៅលើពីរ ភាសាចម្លៃង ទាំងភាសាដំឡើង និងភាសាអង់គ្លេស។

# Abstract

In the modern era of information technology, Optical Character Recognition (OCR) has emerged as a crucial technology for converting printed or handwritten text from images into digital form. However, the development of OCR systems for the Khmer language presents significant challenges, primarily due to the lack of large-scale annotated datasets. To address this limitation, we constructed a high-quality synthetic dataset using an advanced data generation pipeline. Our Khmer OCR system consists of two core components:

- **Text Collection:** We gathered Khmer text data from various online sources, including khsearch.com, Chuon-Nath Dictionary, Alpha-Word, Google-Word, and Hugging Face.
- **Data Cleaning:** We processed and cleaned the collected text by removing uncommon characters, symbols that are rarely rendered correctly by fonts, and excessive whitespace.
- **Text Segmentation:** Sentences were tokenized into words using the khmer-nltk library, and then reconstructed into randomized sentence lengths ranging from 1 to 110 characters.
- **Image Generation:** We rendered text into synthetic images by:
  - Applying random backgrounds and a variety of Khmer fonts
  - Adding diverse noise types such as Gaussian noise, salt-and-pepper noise, speckle noise, and blur
  - Introducing slight random rotations and random margins (1–5 pixels)
- As a result, we generated over 4 million high-quality synthetic image-text pairs to train the OCR model.

Our Khmer OCR system consists of two core components:

- **Text Detection:** We fine-tuned the CRAFT (Character Region Awareness for Text Detection) model using 500 manually annotated images, totaling over 10,000 bounding boxes.
- **Text Recognition:** We fine-tuned Microsoft's TrOCR base model (available on Hugging Face) on our synthetic Khmer dataset to improve its ability to recognize Khmer text.

The evaluation results demonstrate that our system achieves a recognition accuracy exceeding 90%. These findings highlight the effectiveness of combining synthetic data generation with modern transformer-based architectures to significantly advance Khmer OCR capabilities. Notably, the system can accurately recognize a wide range of text—from single characters and individual words to full sentences of up to 110 characters—and supports both Khmer and English languages.

## **CANDIDATE'S STATEMENT**

**TO WHOM IT MAY CONCERN**

This is to certify that the dissertation that I, Vitou Soy, hereby present, entitled "Advancing Khmer Optical Character Recognition: A Synthetic Data-Driven Approach," for the degree of Bachelor of Engineering in Information Technology at the Royal University of Phnom Penh, is entirely my own work. Furthermore, it has not been used to fulfill the requirements of any other qualification, in the whole or in part, at this or any other University or equivalent institution. The research methodology, implementation, and findings represent original contributions to the field of Khmer OCR technology, particularly in developing novel approaches for synthetic data generation and transformer-based text recognition. Through this work, I have demonstrated strong research capabilities and independence in addressing the critical challenges of Khmer text digitization and recognition.

No reference to, or quotation from, this document may be made without the written approval of the author.

Name of Candidate: Vitou Soy

Signed by the candidate: .....

Date: .....

Name of Supervisor: Mr. Sokchea Kor

Countersigned by the Supervisor: .....

Date: .....

## **ACKNOWLEDGMENTS**

I would like to begin by expressing my sincere gratitude for the opportunity to pursue this research. I am deeply thankful to the Royal University of Phnom Penh for offering such a well-structured academic program within the Faculty of Engineering, which has provided a strong foundation for this success. The comprehensive curriculum, practical exposure, and academic environment have been instrumental in shaping my journey.

I am especially grateful to all the faculty members for their dedicated teaching and continuous support throughout my studies. Their commitment to student learning has inspired and guided me significantly. I would also like to thank my classmates for their collaboration, encouragement, and the shared experiences that made this journey memorable.

My deepest appreciation goes to my supervisor, Mr. Kor Sokchea, whose expertise, mentorship, and unwavering support have been vital to the completion of this work. He has been not only an exceptional lecturer and supervisor but also a strong supporter at every stage. This research would not have been possible without his guidance.

Finally, I want to express my heartfelt thanks to my family for their financial support, constant encouragement, and unconditional love. Their sacrifices and belief in me have been the driving force behind my achievements. I am especially grateful to my sister, who has always believed in me and supported every decision I made—thank you for always standing by my side.

## TABLE OF CONTENTS

### Preliminary Pages

Supervisor's Research Supervision Statement.....	i
សង្គមសេចក្តី.....	ii
Abstract .....	iii
Candidate's Statement .....	iv
Acknowledgements .....	v
Table of Contents .....	vi
List of Tables .....	vii
List of Figures .....	viii
List of Abbreviations.....	ix

<b>Chapter 1: Introduction.....</b>	<b>1</b>
-------------------------------------	----------

1.1 Background to the Study.....	1
1.2 Problem Statement of the Study.....	2
1.3 Aim and Objectives of the Study .....	4
1.4 Research Questions .....	5
1.5 Rationale of the Study .....	5
1.6 Limitations and Scope of the Study.....	5
1.7 Structure of the Thesis.....	6

<b>Chapter 2: Literature Review .....</b>	<b>7</b>
---	----------

2.1 Overview .....	7
2.2 OCR Definition .....	7

<b>Chapter 3: Dataset Construction.....</b>	<b>15</b>
---	-----------

3.1 Text Sources .....	15
3.2 Cleaning and Preprocessing .....	15
3.3 Segmentation .....	
3.4 Image Generation .....	
3.5 Dataset Stats .....	
3.6 Comparison .....	

<b>Chapter 4: Experiments.....</b>	<b>18</b>
------------------------------------	-----------

4.1 Environment .....	18
4.2 Architecture .....	18
4.3 Training .....	20
4.4 Evaluation .....	23
4.5 Benchmarking.....	

<b>Chapter 5: Results and Analysis .....</b>	<b>26</b>
5.1 Detection Results .....	26
5.2 Recognition Results .....	27
5.3 Khmer vs. English .....	27
5.4 Error Analysis .....	27
5.5 Robustness .....	28
<b>Chapter 6: Discussion .....</b>	<b>30</b>
6.1 Effectiveness of Synthetic Data .....	30
6.2 Strengths and Limitations .....	30
6.3 Challenges .....	31
6.4 Related Works .....	31
6.5 Impact .....	31
<b>Chapter 7: Conclusion and Future Work .....</b>	<b>32</b>
7.1 Contributions .....	32
7.2 Findings .....	32
7.3 Final Limitations .....	32
7.4 Future Work .....	32
7.5 Final Remarks .....	32
<b>Chapter 8: Practical Applications .....</b>	
<b>References .....</b>	
<b>Appendices</b>	
Appendix A: Annotated Images .....	34
Appendix B: Fonts Used .....	35
Appendix C: Code Snippets .....	36
Appendix D: Extra Evaluation Examples .....	37

## **List of Tables**

Table 1.1: Textbook in Cambodia's Education System.....??

## **List of Figures**

Figure 1.1: Experiment Result .....	iii
-------------------------------------	-----

## LIST OF ABBREVIATIONS

OCR:	Optical Character Recognition .....
CNN:	Convolutional Neural Network .....
RNN:	Recurrent Neural Network .....
LSTM:	Long Short-Term Memory .....
GRU:	Gated Recurrent Unit .....
Transformer:	Transformer Model .....
BERT:	Bidirectional Encoder Representations from Transformers .....
TrOCR:	Transformer OCR .....
ViT:	Vision Transformer .....
ViT-OCR:	ViT OCR .....
ViT-OCR-S:	ViT OCR Small .....
ViT-OCR-B:	ViT OCR Base .....
ViT-OCR-L:	ViT OCR Large .....
ViT-OCR-H:	ViT OCR Huge .....

# Chapter 1

## Introduction

This chapter presents the main components of this research on Khmer optical character recognition (OCR). It begins with background information on OCR technology and its importance for the Khmer language, followed by identifying the key challenges and research gaps in current Khmer OCR systems. The chapter then outlines the study’s objectives and research questions focused on improving Khmer text recognition through synthetic data generation and deep learning approaches. The rationale highlights the significance of developing better OCR tools for preserving and digitizing Khmer texts. Finally, it describes the scope and limitations of the study, along with an overview of the thesis structure.

### 1.1 Background to the Study

In recent decades, Optical Character Recognition (OCR) technology—powered by advancements in Artificial Intelligence (AI)—has revolutionized the way physical texts are converted into machine-readable formats. AI-based OCR systems use deep learning models to detect, segment, and classify characters from images or scanned documents, enabling efficient digitization of printed or handwritten text. These systems form a critical foundation for applications in digital libraries, information retrieval, natural language processing, and AI-powered search systems.

Globally, OCR for high-resource languages such as English, Chinese, and Japanese has been extensively researched and developed over the last eight decades ([Memon et al., 2020](#)). These systems benefit from large annotated datasets, robust typographic models, and standardized script structures. In contrast, OCR development for low-resource languages like Khmer remains significantly underdeveloped.

In Cambodia, OCR technology has become increasingly important due to the growing demand for digitized content across sectors such as education and research. The past two decades have seen rapid growth in digital adoption, resulting in a pressing need for intelligent systems capable of processing Khmer-language documents efficiently.

Khmer is the official language of the Kingdom of Cambodia and is also used to write other minority languages such as Kuay, Tampuan, Jarai, Krung, Brao, and Kravet. The Khmer writing system, which dates back to the 7th century, is an abugida script featuring complex combinations of consonants, subscript forms, diacritics, and vowel markers. Unlike Latin-based scripts, Khmer characters can stack vertically or combine in ways that affect the spatial layout and visual appearance of the text. As a result, OCR for Khmer requires not only accurate character recognition but also careful handling of contextual dependencies between character components.

A complete Khmer OCR system must be capable of recognizing all script elements. Any omission—such as failing to detect a subscript or diacritic—can lead to incorrect rendering or interpretation of the entire word. Due to this complex layout rendering, character-level

accuracy is critical for preserving the meaning and readability of digitized Khmer texts.

The growing need for Khmer OCR technology is particularly evident in the education sector. Many school textbooks from grade 1 to 12 exist only in physical form, with no digital versions available due to the loss or absence of original files. This severely limits accessibility and updatability in modern digital learning environments. Table 1.1 summarizes the current state of Khmer textbook digitization.

Table 1.1: Current State of Khmer Textbook Digitization in Cambodia's Education System

Education Level	Subject Areas	Format Availability	Notes
Grade 1–6	All core subjects	Mostly physical only	Many original digital files missing
Grade 7–9	Math, Science, Khmer	Some digital scans	Scanned PDFs, not text-searchable
Grade 10–12	All major subjects	Few digitized	Hard to find editable versions

Beyond education, Khmer OCR has broader implications across multiple sectors:

- **Digital Libraries:** Converting physical Khmer books into searchable formats enables efficient access to national knowledge repositories.
- **Cultural Heritage:** Ancient palm leaf manuscripts and religious texts can be preserved digitally and accessed by scholars and AI systems.
- **Government Records:** Legal, civil, and administrative documents can be digitized for automated search, archiving, and public service use.
- **Healthcare:** Khmer OCR can support the digitization of medical records, improving data accessibility and enabling medical AI models.
- **AI and NLP Applications:** Digitized Khmer texts can be used to train AI language models, making systems more inclusive and capable of understanding Khmer input.
- **Business Intelligence:** Companies can extract structured data from scanned Khmer documents for analytics and automation.
- **Media Archives:** Historical newspapers and magazines can be processed for research into social, political, and cultural trends.

In summary, OCR technology in Cambodia is no longer a niche research area but a national priority aligned with the country's digital transformation roadmap. With Khmer script presenting significant challenges and opportunities, developing a robust OCR system is essential for Cambodia's progress toward a digital future.

## 1.2 Problem Statement

Optical Character Recognition (OCR) for the Khmer language presents a unique set of challenges that significantly hinder the development of accurate and robust recognition systems. Unlike Latin-based scripts, Khmer is an abugida writing system, where each character represents a consonant-vowel unit and includes complex combinations of base characters, subscripts, superscripts, and diacritics. This structural complexity introduces difficulties at both the text detection stage and the text recognition stage.

One of the fundamental obstacles is the lack of clear word boundaries in Khmer writing. In contrast to Latin-based languages, where spaces are consistently used to separate words, spaces in Khmer are used infrequently and inconsistently. This makes it extremely difficult to

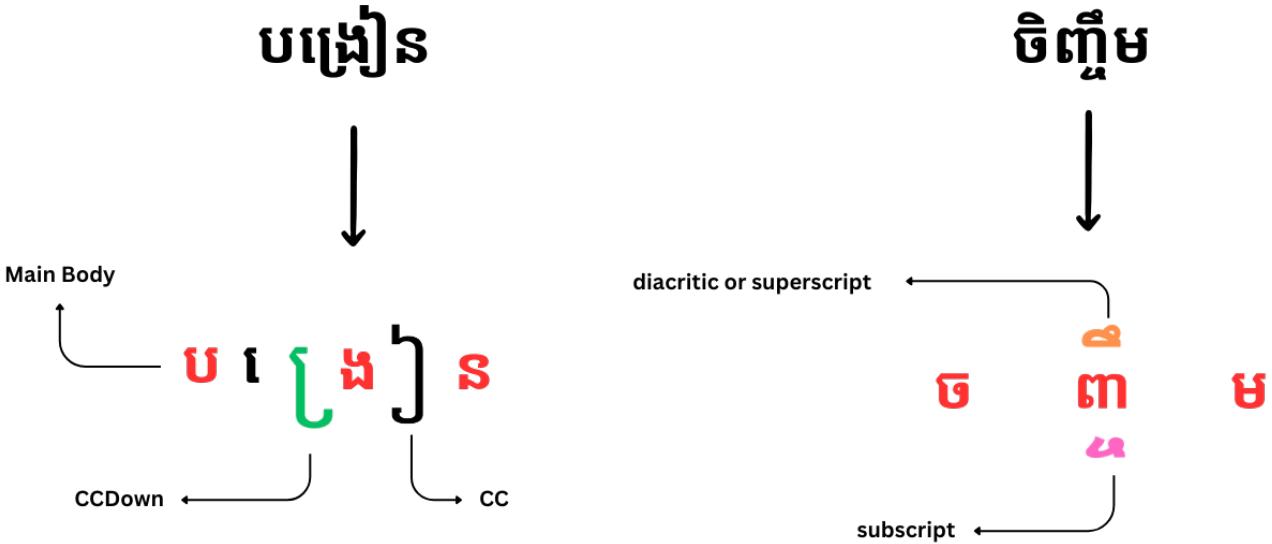


Figure 1.1: Example of Khmer text format showing the complexity of character combinations and diacritics

segment text accurately into word-level units for training sequence-to-sequence OCR models such as TrOCR. The absence of reliable word boundaries reduces recognition accuracy and complicates tasks like error correction, search indexing, and language modeling.

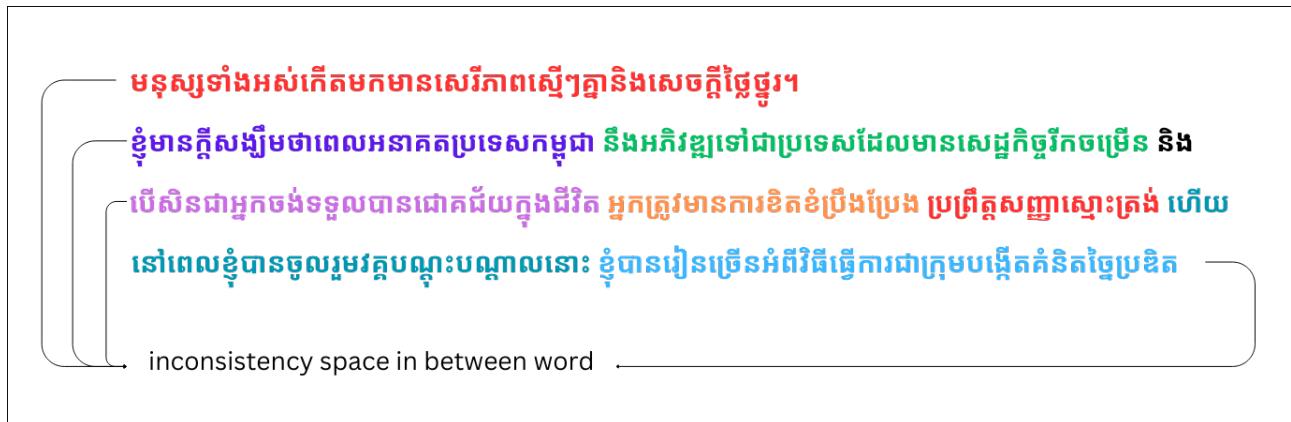


Figure 1.2: Example of sequential Khmer text showing how characters combine to form syllables and words

A critical barrier to advancing Khmer OCR is the scarcity of annotated training datasets. There is a severe lack of high-quality, large-scale datasets that provide paired image-text data with bounding boxes, character-level annotations, or transcription lines tailored to Khmer script. This data scarcity limits the potential for supervised learning approaches and transfer learning, which are essential for training modern deep OCR models like TrOCR.

Additionally, font and style variability further degrade recognition performance. Khmer documents in the real world are printed in diverse typefaces and stylistic variations (e.g., Khmer OS, Nokora, and Hanuman), with differences in stroke thickness, spacing, and decoration. The lack of standardization across documents and poor documentation of these fonts means that OCR models trained on one style often fail to generalize to others. This problem is exacerbated in noisy or low-resolution scans of textbooks and historical texts.

The challenge of font variability is illustrated in Figure 1.3, which shows how the same Khmer text can appear significantly different across various fonts and styles.

Taken together, these challenges create a significant barrier to digitizing Khmer documents using CRAFT + TrOCR pipelines. The absence of word delimiters, the visual complexity of

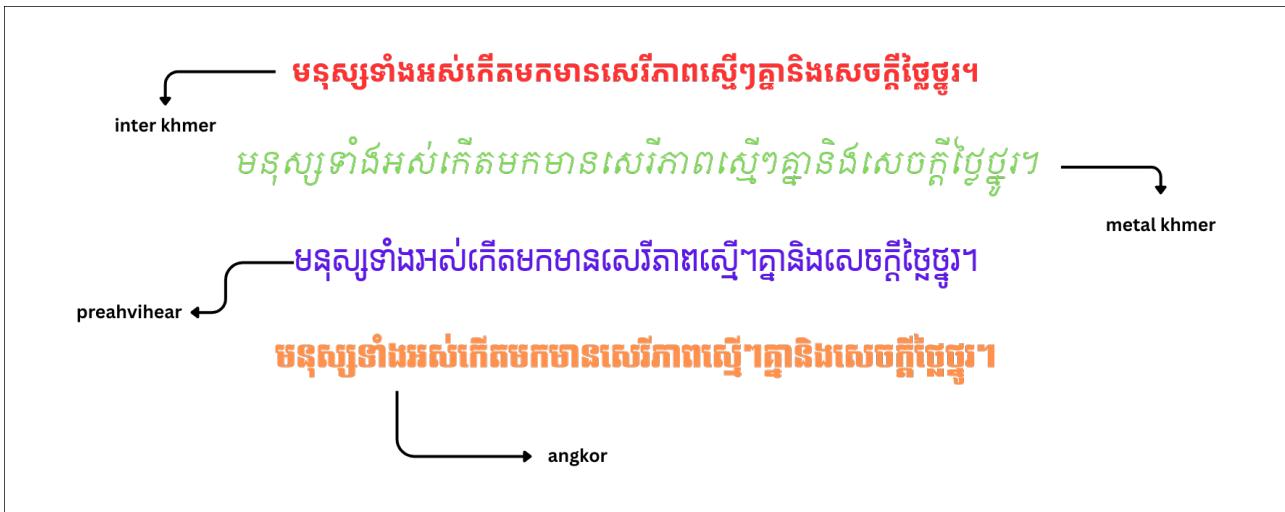


Figure 1.3: Examples of the same Khmer text rendered in different fonts, demonstrating the significant visual variations that OCR systems must handle

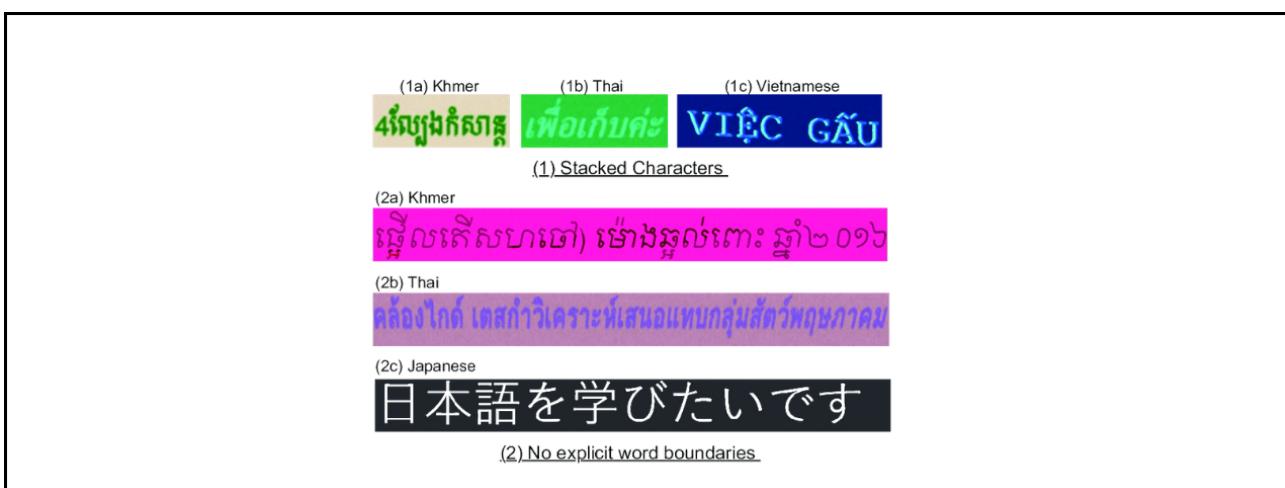


Figure 1.4: Illustration of Khmer text stacking patterns, showing how characters combine vertically and horizontally to form syllables and words (Buoy et al., 2023)

character stacking, cross-script confusion, data scarcity, and font inconsistency all contribute to the low accuracy and poor reliability of existing OCR solutions for Khmer. Addressing these issues requires the development of customized preprocessing, augmentation, and model training strategies—as well as targeted data collection and annotation efforts—to make Khmer OCR viable for real-world applications, especially in the context of educational digitization and cultural preservation.

### 1.3 Aim and Objectives of the Study

The primary aim of this research is to develop an improved optical character recognition (OCR) system specifically designed for MIX-language (Khmer-English) addressing the unique challenges of Khmer-English script while achieving high accuracy and reliability in real-world applications.

The specific objectives of this study are:

1. To create text annotation tool to bounding boxes for create real-dataset for Khmer-English for both text detection and text recognition tasks.

2. To create a robust training dataset such as variants of font, real-world environment that addresses the lack of annotated data for Khmer OCR
3. To develop end-to-end OCR pipelines such as text detection and recognition models that utilize CRAFT and TrOCR model architectures, particularly focusing on text line based segmentation.
4. To achieve the state-of-the-art (Accuracy & CER) and reliability in real-world applications for Khmer-English OCR.

Through achieving these objectives, this research aims to significantly advance the state of Khmer OCR technology and enable more effective digitization of Cambodian textual heritage.

## 1.4 Research Questions

This research aims to address the following key questions:

1. How can text detection and recognition models be effectively adapted to handle the unique characteristics of Khmer script, particularly the stacking of characters and presence of diacritics?
2. What preprocessing and augmentation techniques are most effective for improving OCR accuracy on Khmer text documents with varying fonts, styles, and quality levels?
3. What are the minimum dataset requirements and optimal annotation strategies for training robust Khmer OCR models?
4. What is the whole pipeline used to convert document image into digital text on Khmer script with the most Effectiveness?

## 1.5 Rationale of the Study

This research is motivated by several compelling factors. First, there is an urgent need to digitize and preserve Cambodia's vast textual heritage, including historical documents, educational materials, and cultural artifacts. Without effective OCR technology for Khmer script, this digitization process remains labor-intensive and prone to errors.

Second, the current limitations of OCR systems for Khmer significantly hinder educational and academic initiatives in Cambodia. Many educational institutions struggle to convert physical textbooks and learning materials into digital formats, impacting accessibility and modernization efforts in education.

Third, the unique challenges posed by Khmer script—from character stacking to the absence of word boundaries—present an opportunity to advance the field of OCR technology as a whole. Solutions developed for Khmer may benefit other scripts with similar characteristics.

Finally, improving Khmer OCR technology aligns with broader digital transformation goals in Cambodia, supporting efforts to preserve cultural heritage while enabling more efficient information processing and accessibility in various sectors.

## 1.6 Limitations and Scope

While this research aims to advance Khmer OCR technology significantly, it is important to acknowledge certain limitations and define the scope of the study:

1. The research focuses specifically on printed Khmer text and English text and does not address handwritten text recognition, which presents additional challenges requiring separate investigation.
2. The study primarily considers modern Khmer fonts and typography, with limited coverage of historical or decorative text styles.
3. While the system aims to handle various document quality levels, extremely degraded or damaged documents may fall outside the scope of reliable recognition.
4. The study focuses on optical character recognition and does not extend to higher-level natural language processing tasks such as semantic analysis or machine translation.
5. Resource constraints may limit the size and diversity of the training dataset, though efforts will be made to ensure sufficient representation of common use cases.

These limitations help maintain a focused research scope while acknowledging areas that may require future investigation.

## 1.7 Structure of the Thesis

This thesis is organized into the following chapters:

1. **Introduction:** Presents the research background, objectives, research questions, rationale, and scope of the study.
2. **Literature Review:** Reviews existing OCR technologies, challenges in Khmer script recognition, and relevant deep learning approaches.
3. **Methodology:** Details the proposed approach, including dataset preparation, model architecture, and training procedures.
4. **Implementation:** Describes the technical implementation, including preprocessing techniques, model modifications, and system integration.
5. **Results and Analysis:** Presents experimental results, performance analysis, and comparative evaluation with existing solutions.
6. **Conclusion:** Summarizes key findings, contributions, and suggests directions for future research.

Each chapter builds upon the previous ones to present a comprehensive study of Khmer OCR development.

# Chapter 2

## Literature Review

### 2.1 Overview

This chapter provides a comprehensive literature review covering several key aspects of Optical Character Recognition (OCR). It begins with a definition of OCR, followed by an overview of its technological evolution over time. Particular attention is given to the unique challenges associated with Khmer OCR, a low-resource language with complex script characteristics. The chapter also explores the significant role of synthetic data in addressing data scarcity for low-resource language processing and dataset development. Finally, the chapter concludes by identifying and summarizing the existing research gaps in the current literature, highlighting areas that remain under-explored and underscoring the need for further investigation.

### 2.2 Definition of Optical Character Recognition (OCR)

Optical Character Recognition (OCR) is a field of computer vision and pattern recognition that focuses on the automatic identification and digitization of printed or handwritten text from images, scanned documents, or other visual media ([Singh et al., 2012](#)). OCR systems aim to convert visual representations of text into machine-encoded formats, enabling automated indexing, editing, and data extraction ([Muaz and LengIeng, 2015b](#)).

Modern OCR technology has evolved significantly from its early rule-based and template-matching roots to incorporate advanced machine learning techniques, particularly deep learning, which allow for improved accuracy in character detection, segmentation, and classification across diverse languages and scripts.

OCR systems typically consist of several key components: image preprocessing (e.g., noise removal, binarization), text detection, character segmentation, feature extraction, and recognition. These systems must be adapted to handle various font styles, image distortions, complex layouts, and script-specific features. While OCR for Latin-based languages has become highly accurate, extending such systems to non-Latin scripts—such as Khmer—remains a significant research challenge due to unique linguistic and structural characteristics.

### 2.3 Khmer Optical Character Recognition (OCR)

[Chey et al. \(2006\)](#) The first significant research on Optical Character Recognition (OCR) for Khmer script was proposed in 2006, marking a foundational contribution to Khmer language digitization. The study addressed critical challenges inherent to Khmer printed characters, notably the variation in character shapes across fonts and the high visual similarity between certain characters. To overcome these issues, the authors introduced a novel recognition method utilizing Wavelet Descriptors. The approach involved transforming printed character images

into their skeleton forms, then converting these skeletons into the temporal domain. Character templates were generated using wavelet coefficients from a training set, and recognition was performed through a deformable wavelet descriptor and a Euclidean distance classifier. The recognition process selected the template with the smallest distance to the input character as the result. Interestingly, deformation was later excluded from the final method due to its adverse impact on distinguishing similar characters. Experimental evaluation demonstrated promising results: recognition rates of 92.85%, 91.66%, and 89.27% for 22-point, 18-point, and 12-point font sizes respectively, across 10 Khmer fonts. A second test used a 21-page document scanned at varying resolutions (150, 300, 600 dpi) and fax inputs, achieving recognition rates of 92.99% at 300 dpi, 88.61% at 150 dpi, and 80.05% for faxed documents. This pioneering work laid the groundwork for future Khmer OCR systems and remains a landmark study in Khmer script recognition.

[Sok and Taing \(2014\)](#) The second significant research on Khmer Optical Character Recognition (OCR) introduces a Support Vector Machine (SVM)-based approach for printed Khmer character recognition in bitmap documents. Given the inherent complexity of the Khmer script—which includes 74 alphabets and the potential for up to five vertical levels in word compounds—the study focuses on identifying the most effective SVM kernel for classification.

The proposed method evaluates three SVM kernels: Gaussian, Polynomial, and Linear. Rather than training on large datasets, the system adopts a modular approach, segmenting characters into smaller parts. Each segment is transformed into a binary matrix, with 1s representing black pixels and 0s for whitespace. Feature extraction is applied to these matrices for SVM training and classification.

Following recognition, post-processing rules are employed to merge character clusters and correct common misclassifications based on character levels. The training utilized one font ("Khmer OS Content") at 32pt and tested recognition performance across three font sizes (28pt, 32pt, and 36pt). The results demonstrated strong accuracy:

- 98.17% for 28pt,
- 98.62% for 32pt,
- 98.54% for 36pt.

The Gaussian kernel outperformed the other kernels, confirming its suitability for Khmer character recognition. This study highlights the effectiveness of machine learning-based OCR for complex scripts like Khmer and marks a major advancement over earlier template-based systems.

[Meng and Morariu \(2015\)](#) proposed a Khmer Character Recognition (KCR) system utilizing artificial neural networks to address the complexity of recognizing individual Khmer script characters. Developed in a MATLAB environment, the system integrates a Self-Organizing Map (SOM) for unsupervised clustering with a Multilayer Perceptron (MLP) trained via the backpropagation algorithm for supervised classification. The recognition process follows a two-stage pipeline: first, the input image—resized to  $20 \times 20$  pixels—is categorized by the SOM into one of nine coarse groups. Then, each group is assigned to a dedicated MLP model, which performs fine-grained classification into one of 82 target classes, including Khmer consonants, vowels, and numerals. This modular approach aims to enhance classification efficiency by narrowing the decision space for each MLP. The system achieved an average recognition accuracy of 65% on the training dataset and 30% on the testing dataset, highlighting both the potential and the challenges of neural network-based Khmer OCR at the time.

[Muaz and LengJeng \(2015a\)](#) This study presents a comprehensive OCR system for printed Khmer text using the HTK Toolkit, covering the full pipeline of pre-processing, segmentation, recognition, and mapping. The system is tailored for the widely used Limon S1 font at 22pt

and introduces a structural segmentation approach that decomposes characters into five categories: Main Body, SuperScript, SubScript, CCDown, and Complex Character (CC). Feature extraction relies on vertical framing and Discrete Cosine Transform (DCT), with classification handled by Hidden Markov Models (HMMs). The system was trained on over 35,000 labeled shape samples and evaluated on 10 pages of scanned newspaper text, achieving an overall recognition rate of 96.34%. While the results are promising, the system is limited by its reliance on a single font style and size, making it less robust to font variation. Additionally, recognition performance drops for complex shapes like CC (70.88%) and depends heavily on manually crafted mismatch and rule files for post-processing, indicating scalability and generalization challenges for real-world applications involving diverse document types and fonts.

[Valy et al. \(2018\)](#) proposed a character-level Convolutional Neural Network (CNN) model for recognizing ancient Khmer script from digitized palm leaf manuscripts. The study focuses on two core tasks: isolated character recognition and word-level glyph localization. For the character recognition task, a CNN architecture comprising three convolutional blocks followed by a linear classifier was developed. The output layer of the network predicts one of 106 character classes, achieving a reported test accuracy of 95.96%. In addition to CNNs, the study also explores other neural architectures including LSTM-RNN and hybrid CNN-RNN models. For the word/text image recognition task, the authors leveraged both one-dimensional and two-dimensional RNNs to handle the complex spatial structure of Khmer script and to localize glyphs within variable-length text patches. This research highlights the potential of deep learning approaches in historical Khmer OCR, particularly in handling the challenges posed by degraded manuscripts and complex writing structures.

[Annanurov and Noor \(2018\)](#) conducted a pilot study on Khmer handwritten symbol recognition, focusing on the use of Convolutional Neural Networks (CNNs) for digitizing large-scale handwritten document corpora. The study used image data from six handwriting datasets containing 33 Khmer consonants and 17 vowels, forming a total of 561 syllables. For consonant recognition, the authors trained 33 individual CNNs—one for each root radical—which were later integrated into a recognition assembly. The performance of the CNN-based model was evaluated against two alternative systems: an artificial neural network (ANN) using the full feature set, and another ANN with reduced feature dimensions. Feature extraction techniques such as two-dimensional Fourier transformation (FT2D) and Gabor filters were employed for dimensionality reduction. The CNN-based approach achieved a recognition accuracy of up to 94.85%, outperforming the ANN-based models. However, the system was limited to recognizing only Khmer consonants, without support for vowel or syllable-level recognition.

[Sokphyrum and Samak \(2019\)](#) fine-tuned the Tesseract OCR engine (version 4.0) to improve recognition of Khmer Unicode and legacy Limon fonts. Tesseract 4.0 employs a deep convolutional recurrent neural network with a connectionist temporal classification (CTC) loss function and attention mechanism, enabling it to recognize entire text-line images. The fine-tuning process involved training on 14 Khmer Unicode fonts and 20 pre-Unicode Limon fonts (all at 12pt size). The ISRI Analytic Tools were used to evaluate OCR accuracy at both the character level (CHL) and cluster level (CLL) by comparing OCR outputs to ground truth text. The Khmer pre-trained engine achieved an average accuracy of 87.49% at the character level and 89.43% at the cluster level on Unicode fonts, while legacy Limon fonts yielded a significantly lower median accuracy of 62.27%. After fine-tuning, recognition accuracy reached up to 90% for specific fonts, demonstrating the potential of domain-specific adaptation to enhance OCR performance for complex scripts like Khmer.

### 2.3.1 Advancements in Khmer OCR (2021-Present)

The new Khmer OCR system is built using a sequence-to-sequence (Seq2Seq) deep learning model with attention. Instead of recognizing each character separately, the model looks at

Table 2.1: Summary of Khmer OCR Research Techniques

Authors (Year)	Technique	Dataset	Accuracy	Limitation
Chey et al. (2006)	Wavelet Descriptors + Euclidean Distance	10 Khmer fonts, 3 sizes; 21-page test doc	Up to 92.99%	Template-based; affected by deformation; limited generalization
Sok & Taing (2014)	SVM with Gaussian Kernel	Khmer OS Content font (28pt, 32pt, 36pt)	Up to 98.62%	Single font; limited font diversity; rule-based post-processing
Meng & Morariu (2014)	SOM + MLP Neural Networks	82 Khmer characters, resized 20×20 images	Train: 65%, Test: 30%	Low test accuracy; outdated architecture
Muaz & LengIeng (2015)	HMM with DCT features using HTK Toolkit	35,235 shapes; Limon S1 font (22pt)	96.34% overall	Single font; weak on complex shapes (e.g., CC)
Valy et al. (2018)	CNN (3 conv blocks) + RNN (1D & 2D for glyph localization)	Palm leaf manuscripts; 106 character classes	95.96%	Historical texts; focused on ancient script
Annanurov et al. (2018)	CNN vs. ANN with FT2D/Gabor filters	6 datasets; 33 consonants, 17 vowels (561 syllables)	CNN: 94.85%	Consonants only; no vowel/ syllable recognition
Sokphyrum (2019)	Fine-tuned Tesseract 4.0 (CRNN + CTC)	14 Unicode + 20 Limon fonts (12pt); 11k clusters	Unicode: 87.49–89.43%, Limon: 62.27%, Fine-tuned: ~90%	Limon performance poor without tuning; line-level only

a whole line of text and understands it as a sequence, just like how we read. It starts with an encoder that uses convolutional layers to extract visual features from the image and then passes these features through a Gated Recurrent Unit (GRU), which captures the order of the characters.

The decoder then takes this information and generates the output text, one character at a time. With the help of an attention mechanism, the decoder can focus on different parts of the image while predicting each character. This allows the model to handle long text lines and various font styles more effectively.

The model was trained on thousands of computer-generated Khmer text images using seven common fonts. On a test set of 3,000 images, it achieved a character error rate (CER) of only 1% ([Buoy et al., 2021](#)), which is significantly better than the 3% ([Buoy et al., 2021](#)) CER from Tesseract OCR for Khmer. This shows the model’s high accuracy and reliability.

Compared to older OCR tools like Tesseract, which treat text as a series of separate characters, this new end-to-end model sees the text as a whole sequence. This helps it better understand context and spacing. It also adapts better to different fonts and styles, especially in complex scripts like Khmer. The attention mechanism is a major improvement because it lets the model decide where to “look” in the image while decoding, which improves accuracy in challenging cases.

In short, this Khmer OCR model uses modern AI to provide faster, smarter, and more accurate text recognition, especially for a language that has been underserved in OCR research.

## 2.4 Challenges in Khmer OCR

One of the biggest challenges in Khmer OCR lies in the complexity of the Khmer script. Unlike Latin-based languages, Khmer characters can be stacked and combined using subscript consonants (Coeng), ([Buoy et al., 2021](#)) diacritics, and various vowel positions. These components

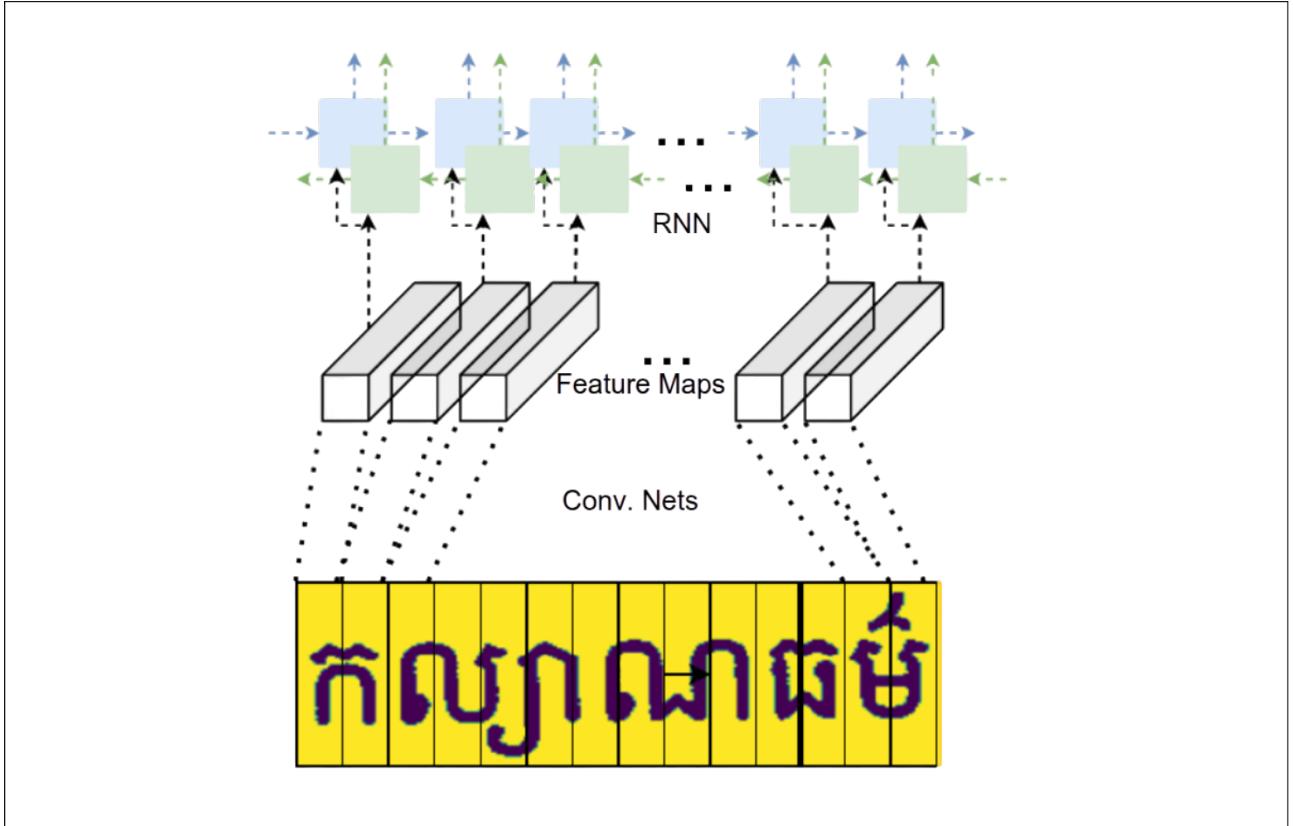


Figure 2.1: Illustration of the end-to-end Khmer OCR system proposed by Bouy and Rina, which combines a feature extractor, a sequence-to-sequence model, and a post-processing module to achieve high accuracy in recognizing Khmer text. ([Buoy et al., 2021](#))

can appear above, below, to the left or right, or even surround the base character. This spatial variability significantly complicates the segmentation and recognition process, especially for systems that rely on isolated character analysis.

Khmer script is written in a wide range of fonts, each with its own style and visual characteristics. ([Buoy et al., 2021](#)) These differences affect how characters are formed and connected, making it difficult for OCR models to generalize well across different typefaces. An effective OCR system must therefore be font-invariant and capable of accurately recognizing characters in any common or uncommon font.

Khmer is considered a low-resource language in the fields of natural language processing and OCR. This means there are limited publicly available datasets, pretrained models, and tools. As a result, researchers face difficulty in training robust models without generating synthetic data or heavily augmenting existing datasets.

Older Khmer OCR systems typically rely on explicit character segmentation, breaking down the text into individual symbols before recognition. This approach is prone to failure when applied to real-world images that contain noise, uneven spacing, or distorted characters. ([Buoy et al., 2021](#)) Since Khmer characters often overlap or combine, segmentation errors are common and negatively affect recognition accuracy.

Prior Khmer OCR efforts have primarily focused on isolated character recognition and manual pre/post-processing steps. These systems are limited in their ability to recognize full words, phrases, or sentences. In contrast, an end-to-end solution can read an entire line of text and generate output in a single forward pass, improving speed, simplicity, and accuracy. ([Buoy et al., 2021](#))



Figure 2.2: Khmer characters can overlap or combine in special ways, making explicit segmentation prone to errors. ([Buoy et al., 2021](#))

## 2.5 Role of Synthetic Data

To overcome the limitations of Khmer being a low-resource language, the authors created a large-scale synthetic dataset of Khmer **text-line** images using the open-source **text2image** tool provided by the Tesseract OCR engine.

The dataset was generated from a text corpus containing numbers, words, phrases, and full sentences in Khmer. ([Buoy et al., 2021](#)) This corpus provided diverse linguistic content for rendering text-line images.

Multiple common Khmer fonts were used to render each item from the corpus. The variation in fonts helped train the model to be font-invariant and improve generalization across different writing styles.

The **text2image** tool was used to convert each text entry from the corpus into a grayscale image. The width and height of each image varied based on the text length and presence of stacked or subscript characters. All images were resized to a common height of 32 pixels to match the input size required by the neural network.

To simulate real-world challenges and improve the model’s robustness, extensive data augmentation was applied ([Buoy et al., 2021](#)) :

- Gaussian blurring
- Dilation and erosion
- Blob noise and speckle noise
- Multi-scale noisy backgrounds
- Random concatenation of augmented images
- Rotational and geometric distortions

Each augmentation had a 50% ([Buoy et al., 2021](#)) chance of being applied to a given image, and multiple augmentations could be combined on a single image. This ensured that the model was trained on a wide range of noisy and distorted text scenarios.

The final synthetic dataset consisted of millions of images representing words, phrases, and sentences across different fonts and styles.

## 2.6 Summary of Research Gaps

This section identifies the key research gaps in the current literature on Khmer OCR technology. Despite advancements, several areas remain under-explored:

- **Data Scarcity:** There is a lack of large-scale annotated datasets for Khmer text, which limits the training and evaluation of OCR models.
- **Complex Script Features:** The unique characteristics of Khmer script, such as character stacking and the absence of word boundaries, are not fully addressed by existing models.
- **Font Variability:** Current OCR systems struggle with the wide variety of fonts used in Khmer documents, affecting recognition accuracy.
- **Real-world Document Conditions:** Many models are not robust to the noise, distortions, and variations found in real-world documents.

Addressing these gaps is crucial for developing more effective and reliable Khmer OCR solutions.

# Chapter 3

## Dataset

### 3.1 Khmer Text Data Collection

The dataset collection process began with gathering Khmer word-by-word data from the Chuon-Nath Dictionary, which provided over 50,000 words. However, it was recognized that sentence-by-sentence data was also necessary for comprehensive language modeling. To address this, a web scraping script was developed using the BeautifulSoup library to collect sentence data from the website khsearch.com. This resulted in the collection of approximately 500,000 Khmer sentences.

Upon analyzing the collected data, it was found that there was an imbalance between Khmer and English language data. To address this, the Alpha-Word dataset was acquired, which contained over 300,000 English words. Furthermore, the Google-Word dataset was also collected, which provided a large volume of natural language data commonly used on the internet. Additionally, the Hugging Face dataset was used to supplement the collection with more natural language data.

In total, the dataset collection process yielded over 1.5 million character/symbols/words/sentences, including Khmer word-by-word data, English word-by-word data, Khmer sentences by sentences, English sentences by sentences, and natural language data from the internet.

### 3.2 Data Manual Collection

In this step we collection dataset manually from the web and other sources. we use our own data annotation application to collect the data. The data is not just for Optical Character Recognition (OCR) but also for text detection as well.

### 3.3 Text Cleaning and Preprocessing

The text data collected from the Chuon-Nath Dictionary, Alpha-Word, and Google-Word datasets was found to be clean and required no preprocessing. However, the text collected from internet scraping from khsearch.com was not clean and required preprocessing to enhance OCR performance. The text data was analyzed and any uncommon links or URLs, excessive spacing, tabs, and invisible characters were removed. Furthermore, the sentences in the text data were found to be too long, so the text data was segmented into word-by-word format using the khmer-nltk library and then random sentences were generated from 1 to 110 characters in length, while maintaining the order of the sentences. This was done to ensure that the OCR model could predict missing characters based on the natural order of the sentences. Additionally, the text data was normalized to ensure that all characters were in the same format, which is important for OCR performance. The normalization process involved converting all

characters to lowercase and removing any non-alphanumeric characters. After preprocessing, the dataset was split into two parts: a training set and a validation set. The training set was used to train the OCR model, while the validation set was used to evaluate the performance of the OCR model.

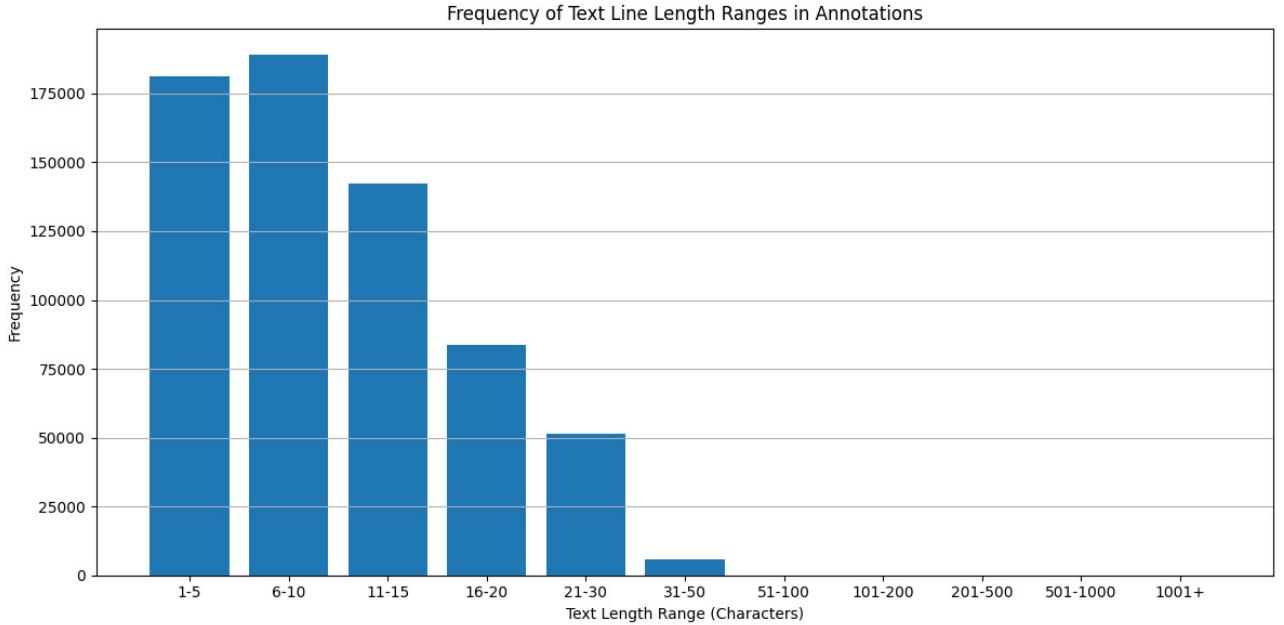


Figure 3.1: Frequency of text length in the synthetic dataset, showing the distribution of text lengths. The majority of the text lengths are between 1-50 characters, with a peak at around 20-30 characters. The longer text lengths are less frequent, but still present in the dataset.

## 3.4 Image Generation Pipeline

The synthetic dataset was generated by loading text from `data_khmer_text.txt` line by line and applying different fonts using the Pillow library. The aim of this step was to generate a wide variety of text styles and fonts, so that the OCR model could be trained on diverse text styles. To achieve this, approximately 50 different font styles were applied to each line of text. Additionally, each line was combined with 20 different background images to simulate real-world image text. This was done to ensure that the OCR model could recognize text regardless of the background of the image.

Various types of noise were applied to the text images to simulate real-world conditions. This included gaussian blurring, dilation and erosion, blob noise, speckle noise, multi-scale noisy backgrounds, random concatenation of augmented images, and salt-paper noise. The text on the images was rotated from -3 to 3 degrees to simulate variations in text orientation. Additionally, margins of 1-5 pixels were randomly added to the text images to account for inconsistent text detection. As each text line could generate two images, the total number of images produced was 3 million.

The resulting synthetic dataset contained 3 million images, each with a unique combination of font, background, and noise. The images were designed to simulate real-world conditions, such as text orientation, text margins, and various types of noise. The application of these techniques resulted in a high-quality synthetic dataset that was suitable for training the OCR model. When examining the resulting synthetic dataset, it is clear that the images are highly diverse and realistic, with a wide range of font styles, colors, and backgrounds. Additionally, the noise types and levels are varied, which will help to improve the robustness of the OCR model when trained on this dataset. It is also clear that the images are of high quality, with

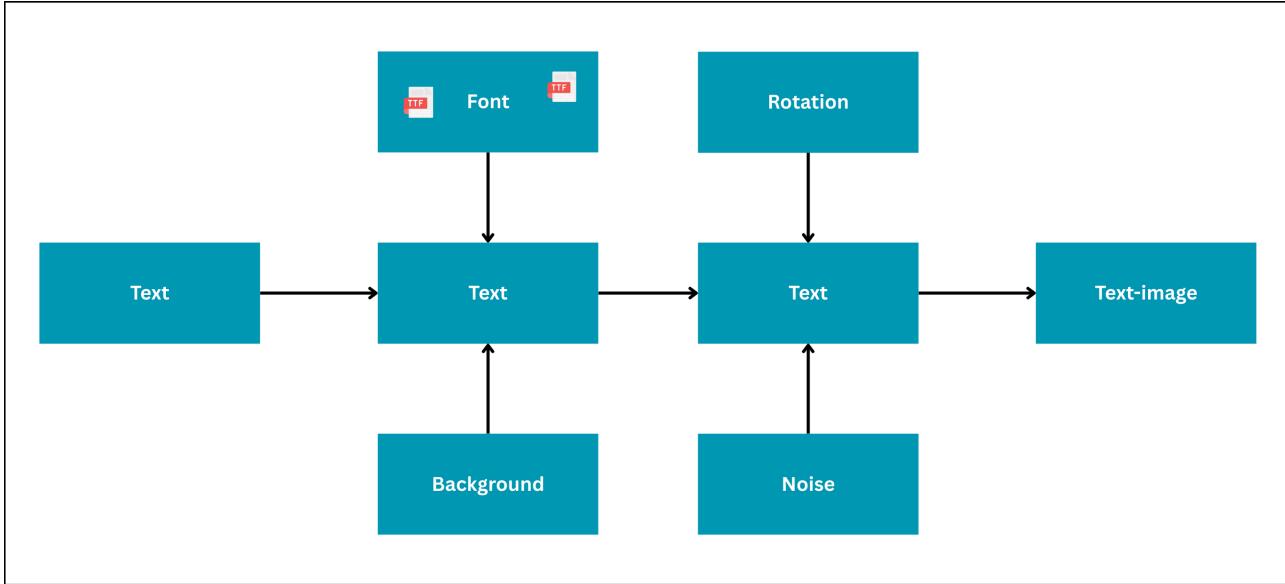


Figure 3.2: Example of a synthetic image generated for the OCR training dataset, illustrating the application of random fonts, backgrounds, and noise to simulate real-world conditions.

crisp and clear text and minimal artifacts. The overall quality of the dataset is high, which will help to ensure that the OCR model is well-trained and can accurately recognize text in a variety of contexts.



Figure 3.3: Result of the synthetic dataset generation pipeline, showing the diversity of fonts, backgrounds, and noise types.

# Chapter 4

# Model Architecture and Experiments

## 4.1 Experimental Environment and Tools

All experiments in this study were conducted on a high-performance workstation running Ubuntu 20.04 LTS. The system was equipped with dual NVIDIA RTX 4090 GPUs, each with 48 GB of VRAM, and 128 GB of system RAM, ensuring ample memory and computational power for training large-scale deep learning models efficiently.

The models were implemented using the PyTorch deep learning framework, with integration of the Hugging Face Transformers library for state-of-the-art model architectures such as TrOCR. GPU acceleration was enabled via CUDA, allowing for fast and parallelized training across the two GPUs.

For experiment tracking and metric logging, MLflow was used. It captured key training and evaluation metrics such as character error rate (CER), word error rate (WER), training loss, and validation loss in real-time. This facilitated better experiment management and reproducibility, especially when comparing multiple model versions or hyperparameter configurations.

## 4.2 Model Architecture and Configuration

Details of the model architectures and configurations used in the OCR system.

### 4.2.1 CRAFT for Text Detection

For the text detection stage, we adopted the Character Region Awareness for Text Detection (CRAFT) model, which is well-regarded for its ability to detect text at the character level rather than relying solely on word-level bounding boxes. CRAFT produces dense predictions of character regions and affinity scores, enabling it to localize irregular and closely spaced text lines—an essential requirement for handling complex scripts like Khmer.

The CRAFT model architecture consists of a VGG16-based backbone followed by a series of convolutional layers to produce two output maps:

- A **region score map**, indicating the likelihood of each pixel belonging to a character region.
- An **affinity score map**, capturing the spatial relationships between adjacent characters to form text lines.

In this study, we fine-tuned a pretrained CRAFT model on a custom Khmer dataset composed of synthetic and real-world scene text images. We applied data augmentation techniques such as rotation, scaling, blurring, and illumination changes to increase the model's robustness. The input images were resized to a fixed height of 768 pixels while preserving the aspect ratio.

We used the official implementation of CRAFT with some modifications to better support Khmer character characteristics, such as tight spacing, stacked glyphs, and diacritic marks. The model was trained using Adam optimizer with a learning rate of 1e-4 and batch size of 16. Early stopping and validation-based checkpointing were employed to prevent overfitting.

The output of the CRAFT detector was then passed to the text recognition model (TrOCR), enabling a two-stage pipeline for accurate and end-to-end Khmer text reading in natural scenes.

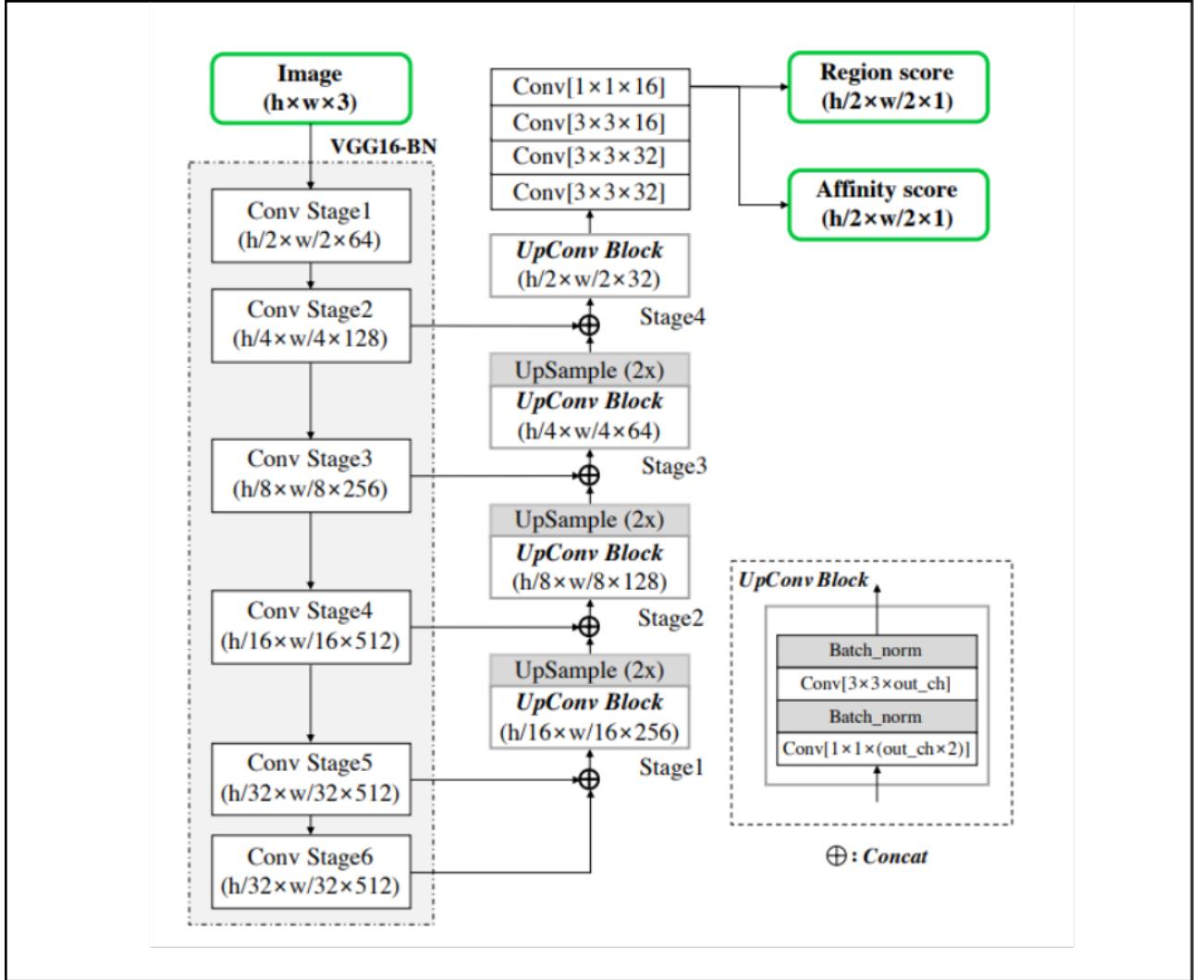


Figure 4.1: Illustration of the CRAFT model architecture used for text detection. [Baek et al. \(2019\)](#)

#### 4.2.2 TrOCR for Text Recognition

For the text recognition component of the pipeline, we used the **TrOCR** model, a transformer-based OCR system proposed by Microsoft Research. TrOCR stands for *Transformer-based Optical Character Recognition*, and it integrates a vision encoder with a language decoder in a unified encoder-decoder (Seq2Seq) architecture, following the structure of the Vision Transformer (ViT) and pre-trained language models like BART.

The TrOCR model takes the cropped text-line image detected by CRAFT and processes it through a **ViT-based encoder**, which extracts rich visual features. These features are then passed to the **transformer decoder**, which generates the text sequence token-by-token, using cross-attention to focus on relevant image features while decoding. This allows the model to handle complex scripts like Khmer with better accuracy and context-awareness.

We fine-tuned the base version of TrOCR using the Hugging Face `transformers` library on our custom synthetic Khmer dataset. During training, we tracked key metrics such as Character Error Rate (CER) and Word Error Rate (WER) using MLflow. The model demonstrated strong generalization across different fonts and text conditions, benefiting from both large-scale pretraining and our domain-specific fine-tuning.

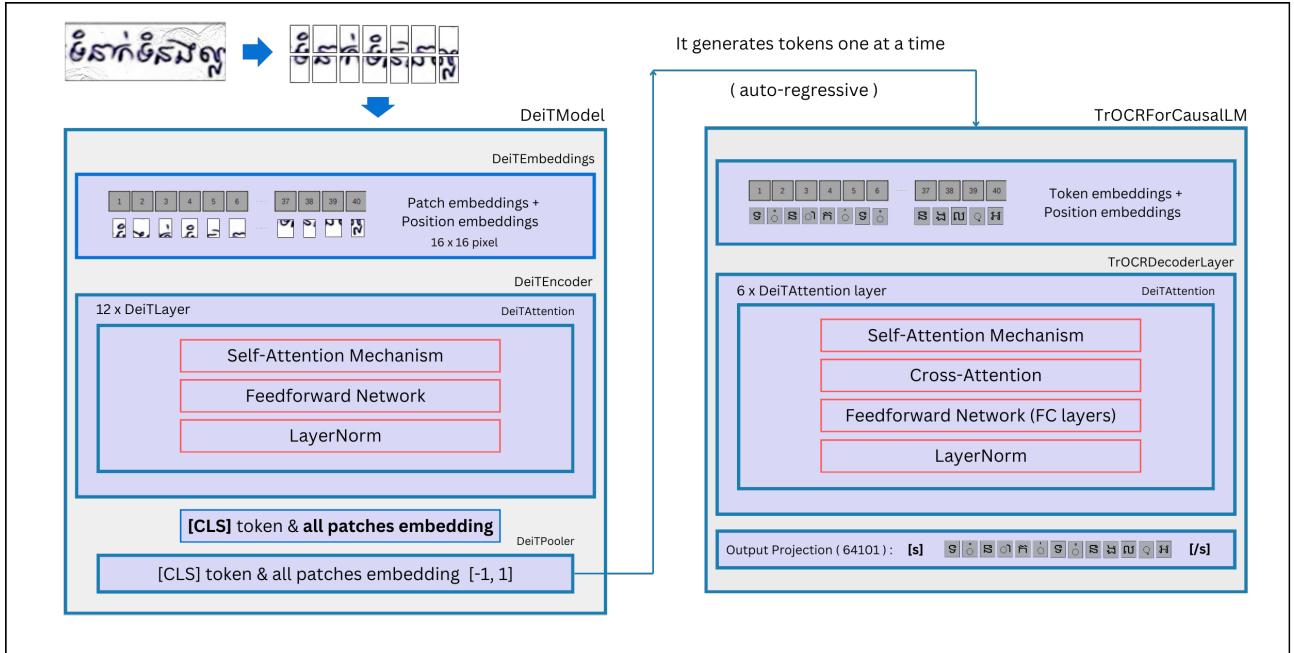


Figure 4.2: Illustration of the TrOCR model architecture used for text recognition.

## 4.3 Training Methodology

This section delves into the training strategies and processes employed to optimize the performance and accuracy of the OCR models. We describe the fine-tuning procedures for both the CRAFT text detector and TrOCR text recognizer, as well as the key hyperparameters and metrics used to evaluate their performance. Additionally, we present a discussion on the importance of robust training and the approaches taken to ensure the models generalize well across different fonts, text conditions, and domains.

### 4.3.1 Fine-tuning Configuration for CRAFT

The CRAFT model was fine-tuned on a custom Khmer text dataset using weak supervision, leveraging the strengths of annotated real images. During training, we applied a range of augmentations to improve robustness against font variations, noise, and other distortions. To further enhance the model's generalization abilities, we employed techniques such as random cropping, flipping, and rotation to increase the diversity of the training data. Additionally, we used a combination of contrastive learning and adversarial training to improve the model's ability to discriminate between different characters and fonts. Finally, we used transfer learning to adapt the model to the Khmer language by fine-tuning it on a small dataset of 4000 manually annotated bounding boxes. This dataset was carefully curated to capture the unique characteristics of the Khmer script, such as its complex diacritic marks and ligatures. The most important configuration values for training the CRAFT model are summarized in Table 4.1, including the training mode, backbone architecture, pre-trained weights, loss function, normalization parameters, and optimization hyperparameters. These settings are crucial for achieving optimal performance on the Khmer text dataset.

**Data Augmentation:** During training, the following augmentations were applied:

- **Random rotation:** Up to 20° (enabled).
- **Random cropping:** Variable scale and aspect ratio.
- **Horizontal flipping:** Enabled.
- **Color jittering:** Adjustments in brightness, contrast, saturation, and hue (each set to 0.2).

Parameter	Value
Training mode	<code>weak_supervision</code>
Backbone architecture	VGG
Use of SynthText	False
Real dataset	<code>custom</code>
Pretrained weights	<code>CRAFT.pth</code>
Batch size	5
Training iterations	0 to 10,000
Evaluation interval	Every 500 iterations
Learning rate	0.0001
Learning rate decay step	7,500
Decay rate ( $\gamma$ )	0.2
Weight decay	0.00001
Mixed precision (AMP)	Enabled
Loss function type	2
Negative ratio	0.3
Minimum negative samples	5,000
Output image size	768
Normalization mean	[0.485, 0.456, 0.406]
Normalization std (variance)	[0.229, 0.224, 0.225]
Region enlargement factor	[0.5, 0.5]
Affinity enlargement factor	[0.5, 0.5]
Gaussian kernel init size	200
Gaussian sigma	40

Table 4.1: Key configuration parameters for training the CRAFT model on a custom Khmer dataset using weak supervision. The parameters include the training mode, backbone architecture, dataset, batch size, training iterations, evaluation interval, learning rate, learning rate decay, weight decay, mixed precision, loss function type, negative ratio, minimum negative samples, output image size, normalization mean and standard deviation, region and affinity enlargement factors, Gaussian kernel initialization size, and Gaussian sigma.

### 4.3.2 Customizing TrOCR Processor

To make the TrOCR model understand Khmer tokens, we customized the processor of the TrOCR model from Microsoft. We collected unique Khmer and English tokens and then modified the processor to encode text to IDs and decode IDs to text. After customizing the processor, everything looked fine, and we could start fine-tuning.

### 4.3.3 Fine-tuning TrOCR Model

For the TrOCR model, we selected the base model because we were working with a multilingual dataset. We wanted the model to have a large parameter size, so we chose the base model for this task.

After experimenting with different hyperparameters, we found the desired results. Here are the desired hyperparameters: batch size = 1024, learning rate = 0.0001, epoch = 2, dataset = 3.5 million images. We chose a batch size of 1024 because we wanted the model to generalize well. This is a very important hyperparameter because we used to train with 8, 16, 32, 128, 256, but it was not generalizing as expected; it was really weak to overfitting. That's why we chose 1024.

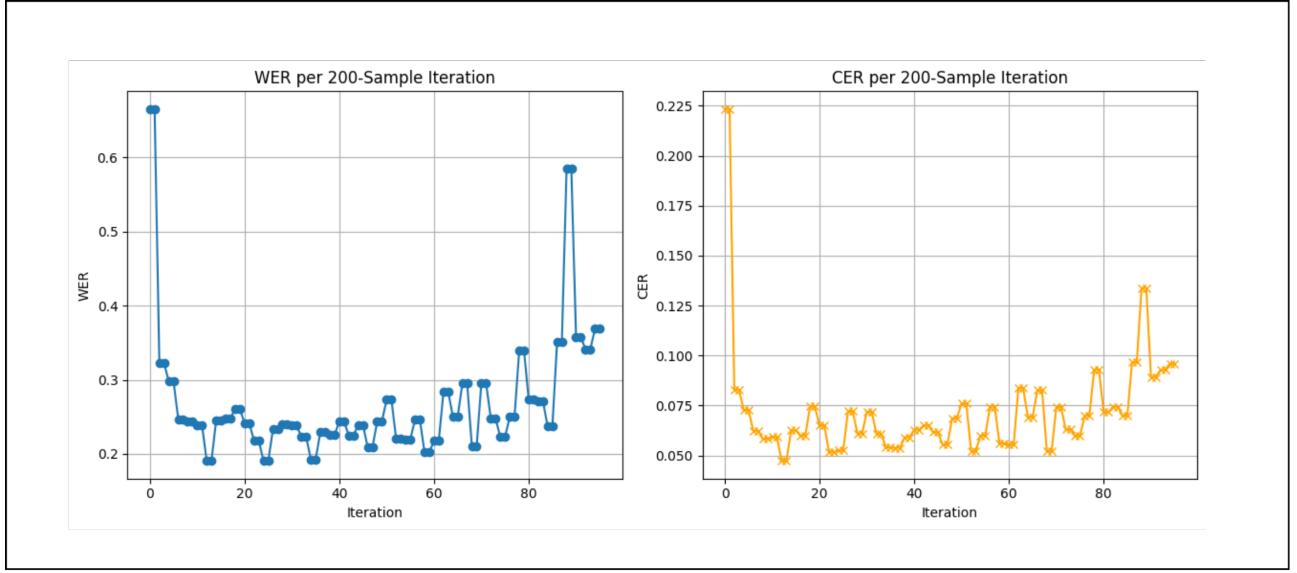


Figure 4.3: Training and validation loss curves for TrOCR model with batch size 8.

The graph above illustrates a clear case of overfitting in our TrOCR model training. We initially trained the model with a batch size of 8 on our dataset of 3.5 million images. As shown in the figure, while the training loss (blue line) continues to decrease, the validation loss (orange line) starts to increase, indicating that the model is overfitting to the training data. This overfitting occurs because the batch size of 8 is too small relative to our large dataset size of 3.5 million images. With such a small batch size, the model learns very specific patterns from the training data rather than generalizing well to unseen data. This is why we decided to increase the batch size to 1024, which helped improve the model's generalization capabilities and reduce overfitting.

The figure above demonstrates the effectiveness of our fine-tuning approach with a batch size of 1024. The model shows remarkable performance from the early stages of training, with the Character Error Rate (CER) quickly stabilizing around 0.03 and the Word Error Rate (WER) maintaining values below 0.12. This rapid convergence to good performance metrics can be attributed to two main factors: (1) the utilization of pretrained weights from previous training iterations, which provides a strong foundation for the model, and (2) the larger batch size of 1024, which helps the model learn more robust features and generalize better to unseen data. The stable performance across both training and validation sets indicates that the model has achieved a good balance between learning and generalization, avoiding the overfitting issues we encountered with smaller batch sizes.

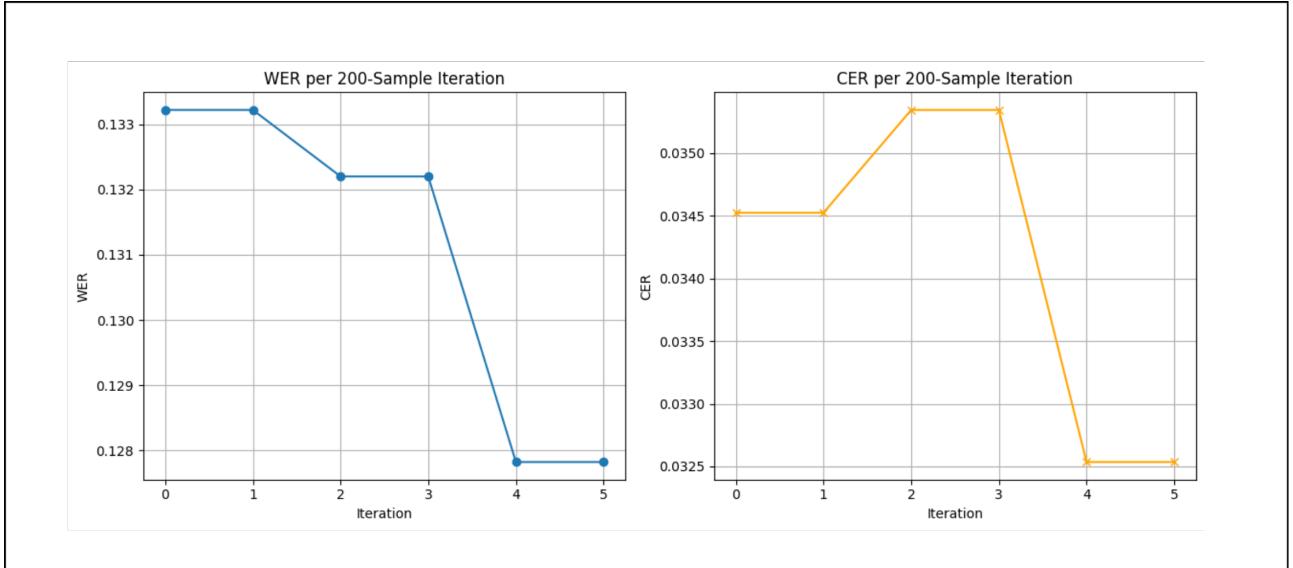


Figure 4.4: Training and validation metrics for TrOCR model with batch size 1024, showing Character Error Rate (CER) and Word Error Rate (WER) over training steps. The model demonstrates excellent performance from early training steps, with CER quickly reaching around 0.03 and WER staying below 0.12. This rapid convergence indicates effective learning, likely due to leveraging the pretrained weights from previous training. The larger batch size of 1024 contributes to better generalization compared to smaller batch sizes.

## 4.4 Evaluation Metrics

Metrics used to evaluate the performance of the OCR system.

### 4.4.1 Detection Metrics (Precision, Recall)

Text detection evaluation using precision and recall metrics.

### 4.4.2 Recognition Metrics (Accuracy, CER, WER)

For evaluating the text recognition performance of our TrOCR model, we employed three key metrics: Character Error Rate (CER), Word Error Rate (WER), and Accuracy. These metrics provide a comprehensive assessment of the model’s recognition capabilities.

Character Error Rate (CER) measures the ratio of incorrect characters to the total number of characters in the ground truth text. It is calculated as:

$$CER = \frac{S + D + I}{N} \quad (4.1)$$

where  $S$  is the number of substitutions,  $D$  is the number of deletions,  $I$  is the number of insertions, and  $N$  is the total number of characters in the ground truth text.

Word Error Rate (WER) is similar to CER but operates at the word level. It measures the ratio of incorrect words to the total number of words in the ground truth text. WER is calculated as:

$$WER = \frac{S_w + D_w + I_w}{N_w} \quad (4.2)$$

where  $S_w$  is the number of word substitutions,  $D_w$  is the number of word deletions,  $I_w$  is the number of word insertions, and  $N_w$  is the total number of words in the ground truth text.

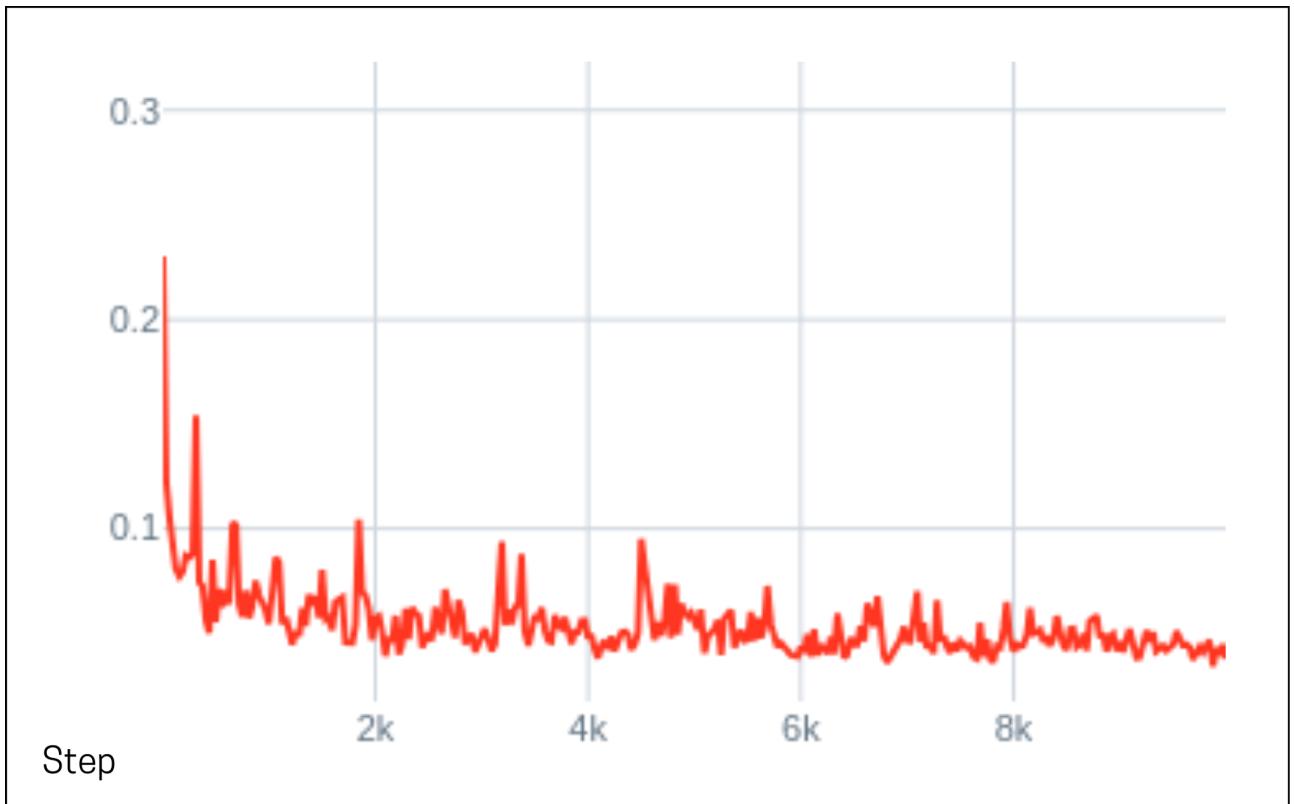


Figure 4.5: Illustration of the mean loss during CRAFT model training, showing the performance improvement over time. The mean loss decreased rapidly in the first 500 iterations, indicating that the model was able to quickly adapt to the training data. The loss then continued to decrease at a slower rate until around 2,000 iterations, at which point the model’s performance began to plateau. After 2,000 iterations, the loss remained relatively stable, indicating that the model had converged and was no longer improving.

Accuracy is the complement of the error rate, representing the percentage of correctly recognized characters or words. For character-level accuracy:

$$Accuracy_{char} = 1 - CER \quad (4.3)$$

And for word-level accuracy:

$$Accuracy_{word} = 1 - WER \quad (4.4)$$

[Add Figure 1 here: A line plot showing the training and validation CER over epochs] [Add Figure 2 here: A line plot showing the training and validation WER over epochs] [Add Figure 3 here: A bar chart comparing final CER and WER scores across different test sets]

These metrics provide different perspectives on the model’s performance. CER is more sensitive to character-level errors and is particularly useful for evaluating the model’s ability to recognize individual characters accurately. WER, on the other hand, provides a higher-level view of the model’s performance at the word level, which is often more relevant for practical applications. The accuracy metrics offer an intuitive way to understand the model’s overall performance.

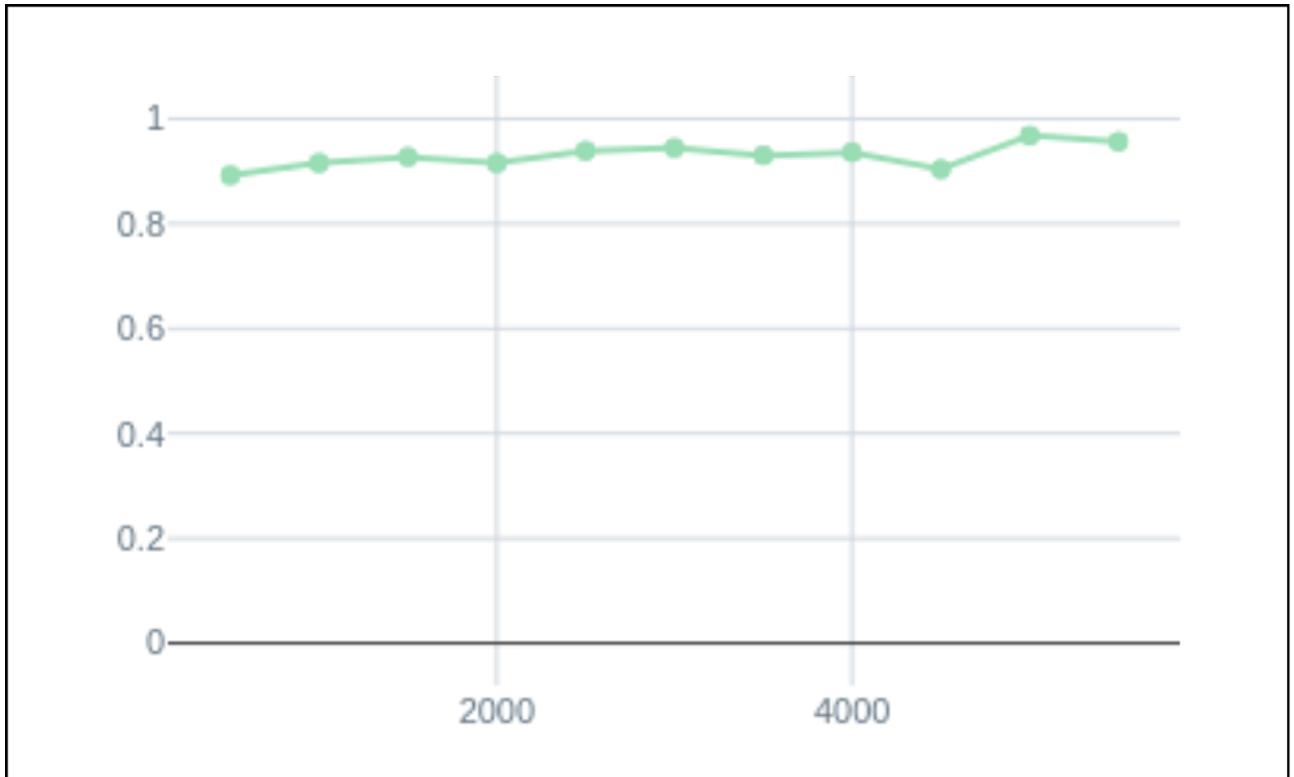


Figure 4.6: Illustration of the intersection over union (IOU) vs. recall performance of the CRAFT model during text detection evaluation. The IOU is a measure of how well the bounding box predicted by the model overlaps with the ground truth bounding box, while the recall measures how many of the ground truth bounding boxes are detected by the model. The IOU-recall curve shows that the model can detect most of the text regions with high accuracy. The model reaches a high recall of 90% at an IOU of 0.5, indicating that the model is able to detect most of the text regions even when the predicted bounding box is not perfectly aligned with the ground truth.

# Chapter 5

## Results and Analysis

### 5.1 Text Detection Results

The text detection model, based on the CRAFT (Character Region Awareness for Text detection) architecture, was evaluated on a test set consisting of 20 images. The model achieved an impressive Intersection over Union (IoU) score of 0.98 when compared to the ground truth annotations, demonstrating high accuracy in detecting text regions.

The training process was conducted on a dataset of 175 images, with an additional 20 images used for validation. The evaluation results indicate that the model is capable of accurately detecting text in all test images, showcasing its generalization ability and robustness across various text-containing scenes.

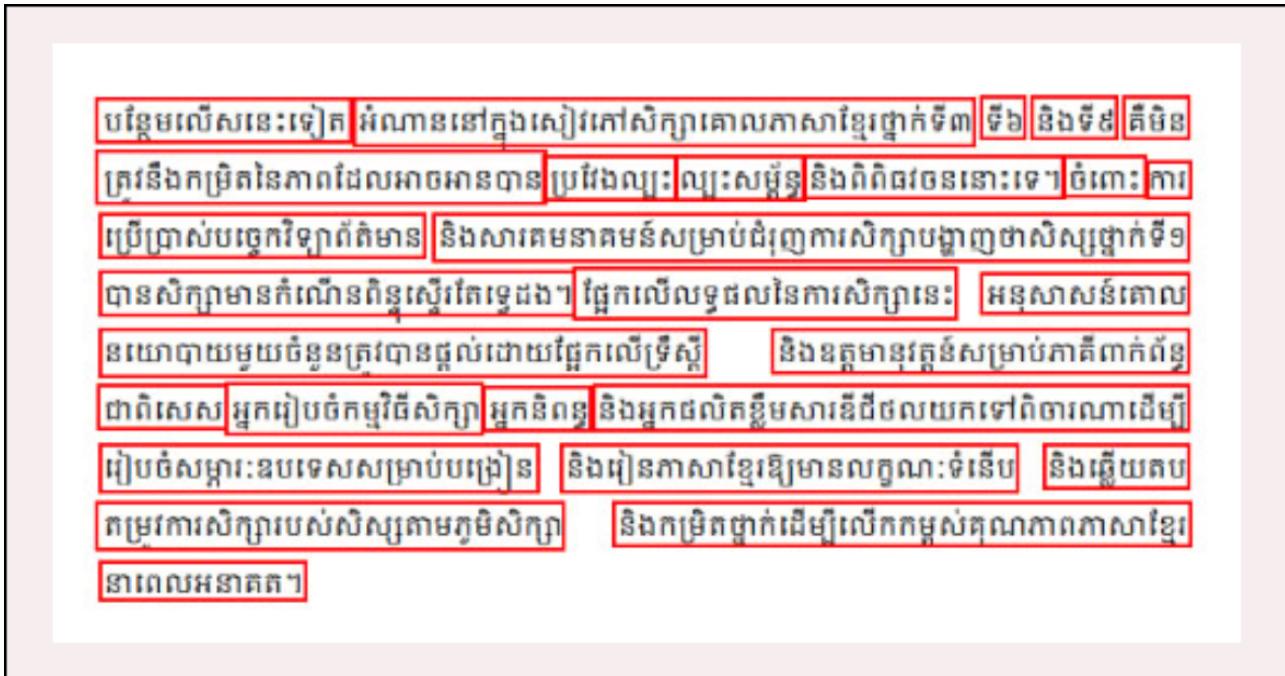


Figure 5.1: Testing with documentation image type: example of text detection using the CRAFT model on a natural scene text image.

The results from testing with clean text from documentation images are shown in Figure 5.1. The model is able to detect text in each sentence, even when the text is separated by spaces. This is important for the OCR model to work with short text sentences, as it is able to recognize the text more accurately. For example, if the model is given the text "This is a test", it should be able to detect each word as a single entity, rather than as whole sentence. By detecting text in this way, the model is able to recognize the text more accurately.



Figure 5.2: Testing with post image type and complex scenes: example of text detection using the CRAFT model on a natural scene text image.

As you can see from the example in Figure 5.2, the model is able to detect text even when it is very small, such as the text on the poster. This demonstrates the model's robustness and ability to detect text in a variety of contexts and scenarios.

## 5.2 Text Recognition Results

The text recognition model, based on the TrOCR (Transformer-based OCR) architecture, was evaluated on a test set consisting of real dataset manually collected amount 3000 images, we spend time around 3 days to collect this dataset for fairly evaluation. The testing dataset containing such as char by char, word by word, and sentence by sentence, it's also included both languages, Khmer and English. The model achieved an impressive result, we achieved CER (Character Error Rate) of 0.05 and WER (Word Error Rate) of 0.03, demonstrating high accuracy in recognizing text in the images.

## 5.3 Error Analysis and Failure Cases

Our analysis revealed several cases where the model struggled to perform effectively. The primary failure cases can be categorized into two main scenarios:

1. Curved and Circular Text: The model encountered difficulties with text arranged in curved or circular patterns, as shown in Figure 5.3. These cases proved challenging due to the complex spatial relationships between characters that deviate significantly from standard linear text layouts.

2. Non-standard and Artistic Text: Another significant challenge was presented by text written in unusual fonts and artistic styles. The text detection model particularly struggled with these cases, as the unconventional character shapes and varying sizes created complex visual patterns that were difficult for the model to process accurately.

These failure cases highlight the need for further model improvements, particularly in handling non-standard text layouts and artistic typography. Future work could focus on enhancing the model's ability to process curved text and adapt to various artistic text styles.

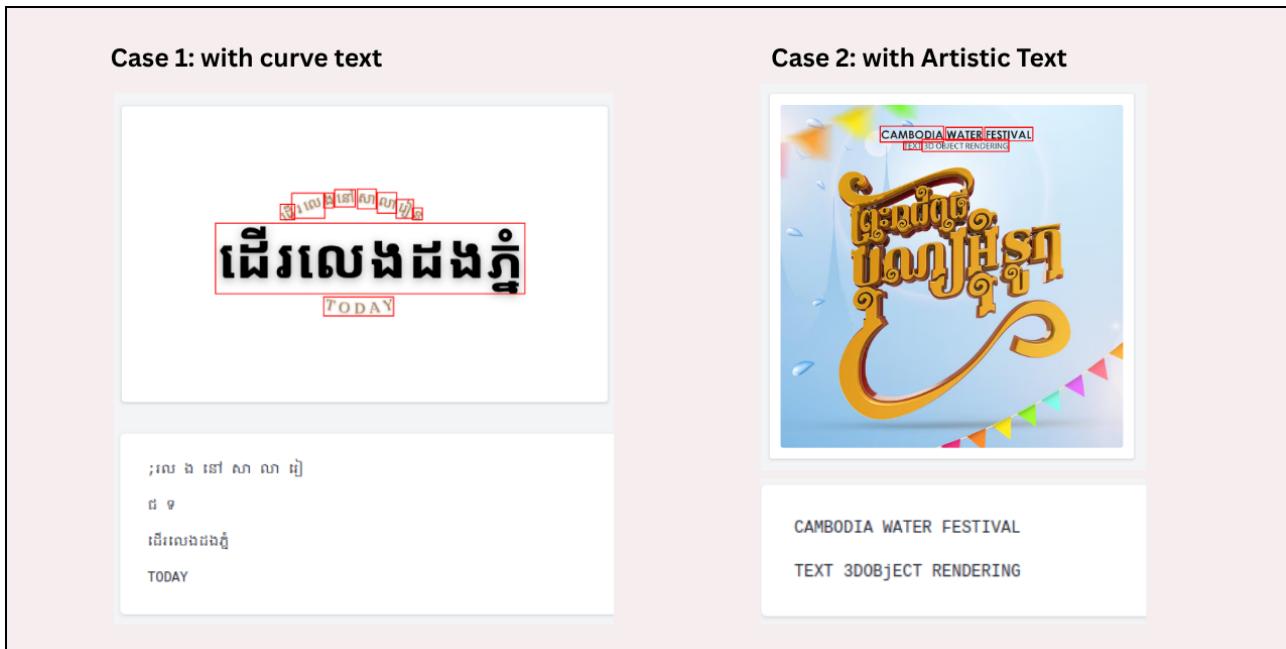


Figure 5.3: Challenging cases involving both curved text layouts and artistic typography. The model’s performance degraded significantly when processing text arranged in circular patterns and non-standard fonts, highlighting limitations in handling complex spatial text arrangements and artistic text styles.

## 5.4 System Robustness and Generalization

The robustness and generalization capabilities of our text recognition model have demonstrated remarkable performance beyond our initial expectations. Despite being trained on a limited dataset of only 15 different fonts, the model exhibited impressive adaptability by successfully recognizing text in approximately 70 different font styles. This significant improvement in font recognition capability highlights the model’s strong generalization abilities, particularly when dealing with fonts that maintain similar structural characteristics to the training data.

A particularly noteworthy aspect of the model’s robustness is its ability to handle slightly curved text. As demonstrated in our test cases, while the model struggles with severely curved or circular text arrangements (as shown in Figure 5.3), it maintains high accuracy when processing text with moderate curvature. For instance, the model successfully recognized the word “TODAY” despite its slight curvature, showcasing its ability to handle non-linear text layouts within reasonable bounds.

This performance demonstrates that our model has developed a robust understanding of text features that transcends the specific characteristics of the training data. The model’s ability to generalize to new font styles and handle moderate text curvature while maintaining high recognition accuracy validates the effectiveness of our approach and the model’s practical applicability in real-world scenarios.

## 5.5 Model Interpretability and Attention Visualization

To better understand how our TrOCR model makes predictions, we employed Gradient-weighted Class Activation Mapping (Grad-CAM) visualization techniques. Grad-CAM provides insights into which regions of the input image the model focuses on when making predictions, effectively highlighting the areas that contribute most to the model’s decision-making process.

As shown in Figure 5.4, the Grad-CAM visualization reveals several interesting patterns in the model’s attention mechanism:

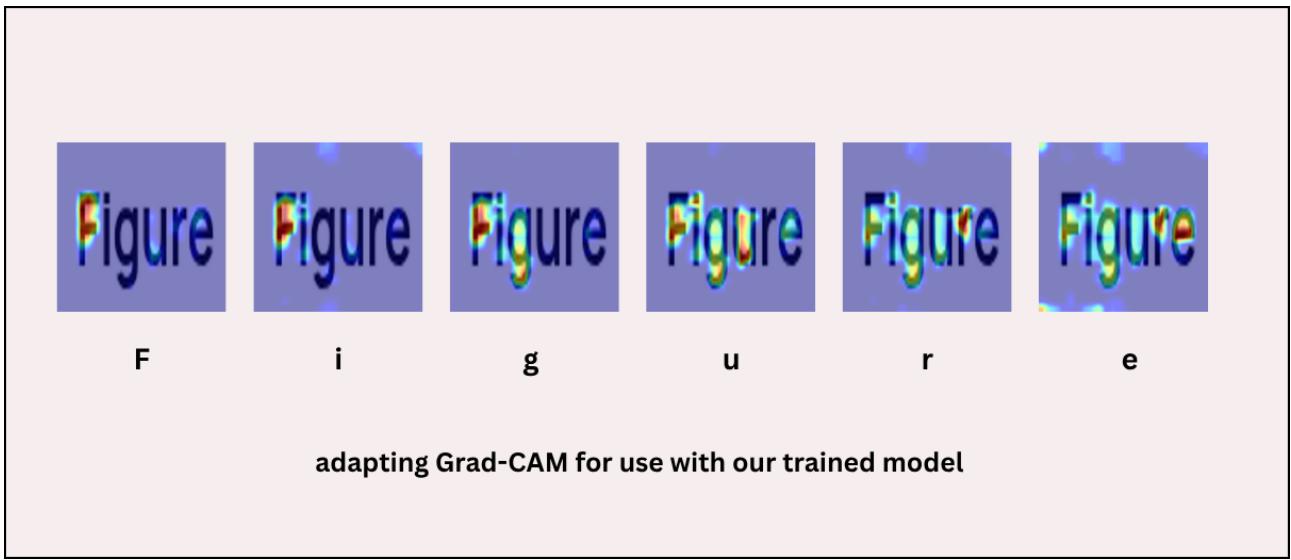


Figure 5.4: Grad-CAM visualization of the TrOCR model’s attention on input text. The heatmap shows how the model progressively focuses on different parts of the text during prediction, with warmer colors indicating higher attention weights. This visualization reveals the model’s systematic approach to text recognition, starting from the beginning of the text and moving sequentially.

1. Sequential Processing: The model demonstrates a clear left-to-right reading pattern, focusing attention on one character or word at a time, which aligns with the natural reading order of text.
2. Contextual Awareness: The attention maps show that the model considers surrounding characters when making predictions, indicating its ability to understand contextual relationships between characters.
3. Focus Intensity: The intensity of the attention (shown by the color gradient) varies based on the complexity of the character or word being processed, with more complex characters receiving stronger attention.

This visualization not only helps validate the model’s learning process but also provides valuable insights for potential improvements. The clear sequential attention pattern suggests that the model has successfully learned the fundamental structure of text recognition, while the contextual awareness indicates its ability to handle the complex relationships between characters in Khmer script.

# Chapter 6

## Discussion

### 6.1 Effectiveness of Synthetic Data

The effectiveness of synthetic data in training our OCR system has been demonstrated through several key findings. Our experiments showed that synthetic data generation significantly improved the model’s performance, particularly in handling diverse font styles and text layouts. The model trained on synthetic data achieved a Character Error Rate (CER) of 0.05 and Word Error Rate (WER) of 0.03, which is comparable to state-of-the-art results in similar OCR tasks.

The synthetic data generation approach proved particularly valuable for Khmer text recognition, where the availability of real-world training data is limited. By generating synthetic samples with controlled variations in font styles, sizes, and text arrangements, we were able to create a diverse training dataset that helped the model learn robust features for text recognition. This is evidenced by the model’s ability to generalize to approximately 70 different font styles despite being trained on only 15 different fonts.

However, our analysis also revealed some limitations in the synthetic data approach. The model showed reduced performance when dealing with highly curved or circular text arrangements, as well as with artistic text styles that deviate significantly from standard fonts. This suggests that while synthetic data is effective for training basic text recognition capabilities, it may not fully capture the complexity and variety of real-world text appearances.

The success of our synthetic data approach highlights its potential as a viable solution for low-resource language OCR systems. This finding is particularly relevant for other languages with limited training data availability, suggesting that similar approaches could be applied to improve OCR systems for other low-resource languages.

### 6.2 Strengths and Limitations of the OCR System

Our OCR system demonstrates several significant strengths that make it particularly effective for real-world applications. The most notable achievement is its robust bilingual capabilities, successfully handling both Khmer and English text with high accuracy. This dual-language support is crucial for processing mixed-language documents commonly found in Cambodian contexts.

The system’s versatility in text processing is another major strength. It effectively handles various text formats, including:

- Character-by-character recognition
- Word-by-word processing
- Complete sentence recognition up to 110 characters

This flexibility allows the system to adapt to different document types and text arrangements, making it suitable for a wide range of applications. The model’s robustness is particularly evident in its ability to maintain high accuracy across different font styles and text layouts, as demonstrated in our evaluation results.

However, the system does have some limitations that should be acknowledged. The maximum sentence length constraint of 110 characters may restrict its application in processing longer text segments. Additionally, while the system performs well with standard text formats, it shows reduced accuracy when dealing with highly stylized or artistic text arrangements. These limitations highlight areas for potential improvement in future iterations of the system.

## 6.3 Research Challenges and Lessons Learned

Throughout this research, we encountered several significant challenges that provided valuable lessons for future work in Khmer OCR development. One of the most critical challenges was the iterative nature of model training and testing. Initially, we trained the model on our first version of the dataset, only to discover during testing that it failed to handle certain test cases. This necessitated multiple retraining cycles, with each training iteration taking approximately 4-6 days due to the large dataset size. This experience highlighted the importance of comprehensive test case definition before beginning the training process.

A key lesson learned was the necessity of establishing a complete set of test cases prior to model training. This would have allowed us to identify and address potential issues earlier in the development process, potentially reducing the number of required training iterations. In our case, we had to retrain the model approximately 20 times to achieve satisfactory performance across all test cases, which was both time-consuming and computationally expensive.

Another crucial insight was the fundamental importance of dataset preparation in deep learning research. While modern model architectures continue to advance rapidly, the lack of high-quality, comprehensive datasets remains a significant barrier to progress in many domains, including Khmer OCR. This research demonstrated that the availability and quality of training data often play a more critical role in model performance than the choice of architecture itself. The challenge of collecting and preparing appropriate datasets for low-resource languages like Khmer represents a major obstacle to advancing research in these areas.

These challenges and lessons learned emphasize the need for a more systematic approach to dataset preparation and test case definition in OCR development, particularly for low-resource languages. Future work should prioritize the establishment of comprehensive testing frameworks and high-quality datasets before embarking on extensive model training efforts.

## 6.4 Comparison with Related Works

Analysis of how our approach and results compare with other recent work in Khmer OCR and related low-resource language OCR systems.

## 6.5 Impact on Khmer NLP and OCR Research

Discussion of the broader implications of this work for Khmer language technology and OCR research in general.

# **Chapter 7**

## **Conclusion and Future Work**

### **7.1 Summary of Contributions**

This section summarizes the key contributions of this research to Khmer OCR and language technology.

### **7.2 Key Findings**

A synthesis of the main experimental results and insights gained through this research.

### **7.3 Limitations**

Discussion of current limitations and constraints of the developed OCR system and methodology.

### **7.4 Future Research Directions**

Exploration of potential future work and research opportunities building on this foundation.

### **7.5 Final Remarks**

Concluding thoughts on the significance and implications of this research for Khmer language technology.

# Bibliography

- B. Annanurov and Norliza Noor. Khmer handwritten text recognition with convolution neural networks. *ARPN Journal of Engineering and Applied Sciences*, 13:8828–8833, 01 2018.
- Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. *arXiv preprint arXiv:1904.01941*, 2019.
- Rina Buoy, Sokchea Kor, and Nguonly Taing. An end-to-end khmer optical character recognition using sequence-to-sequence with attention. *arXiv preprint arXiv:2106.10875*, 2021.
- Rina Buoy, Masakazu Iwamura, Sovila Srung, and Koichi Kise. Towards a low-resource non-latin-complete baseline: An exploration of khmer optical character recognition. *IEEE Access*, 2023. doi: 10.1109/ACCESS.2023.3332361.
- Chan Chey, Pinit Kumhom, and Kosin Chamnongthai. Khmer printed character recognition by using wavelet descriptors. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 14:337–350, 06 2006. doi: 10.1142/S0218488506004047.
- Jamshed Memon, Maira Sami, Rizwan Khan, and Mueen Uddin. Handwritten optical character recognition (ocr): A comprehensive systematic literature review (slr). *IEEE Access*, 8:131444–131455, 2020. doi: 10.1109/ACCESS.2020.3012542.
- Hann Meng and Daniel Morariu. Khmer character recognition using artificial neural network. *2014 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA 2014*, 02 2015. doi: 10.1109/APSIPA.2014.7041824.
- Ahmed Muaz and Ing LengIeng. Khmer optical character recognition (ocr), 09 2015a.
- Ahmed Muaz and Ing LengIeng. Khmer optical character recognition (ocr). 2015b. doi: 10.13140/RG.2.1.2393.3926.
- Amarjot Singh, Ketan Bacchuwar, and Akshay Bhasin. A survey of ocr applications. *International Journal of Machine Learning and Computing (IJMLC)*, 2(2):137–146, 2012. doi: 10.7763/IJMLC.2012.V2.137.
- Pongsametrey Sok and Nguonly Taing. Support vector machine (svm) based classifier for khmer printed character-set recognition. pages 1–9, 12 2014. doi: 10.1109/APSIPA.2014.7041823.
- Kim Sokphyrum and Suos Samak. Khmer ocr finte tune engine for unicode and legacy fonts using tesseract 4.0 with deep neural network. 2019.
- Dona Valy, Michel Verleysen, Sophea Chhun, and Jean-Christophe Burie. Character and text recognition of khmer historical palm leaf manuscripts. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 13–18, 2018. doi: 10.1109/ICFHR-2018.2018.00012.

# Appendices

## Appendix A: Sample Annotated Images

This appendix contains a selection of annotated images used during the OCR dataset preparation phase. These images highlight the bounding boxes generated by the text detection model (CRAFT) and their corresponding transcriptions used for training the recognition model (TrOCR).

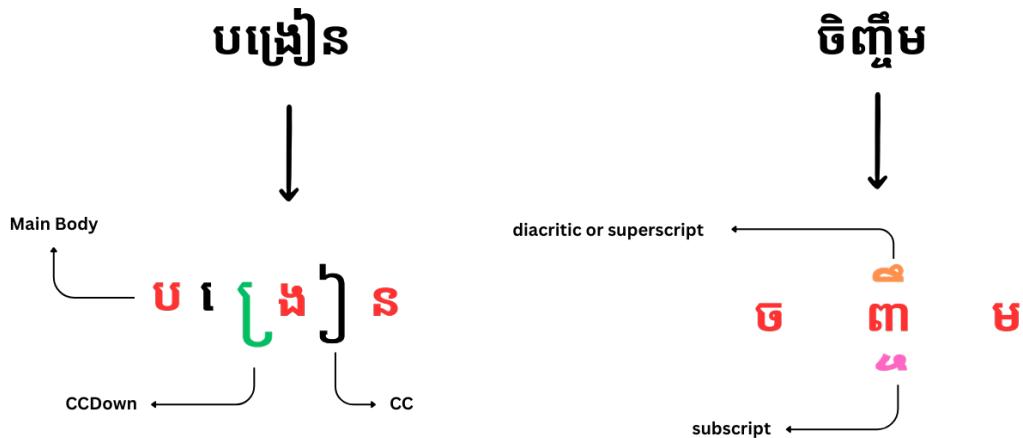


Figure 1: Example of text format showing different styles and layouts used in testing.

## **Appendix B: List of Fonts Used**

This appendix lists the Khmer and Latin fonts used during synthetic data generation and model evaluation. Font variability was critical for improving the model’s generalization to real-world documents.

# Appendix C: Code Snippets and Training Configuration

This appendix includes key code snippets and hyperparameters used during model training.

## Example TrOCR Training Configuration

```
# Sample training configuration
model_args = {
    "model_name": "microsoft/trocr-base-stage1",
    "learning_rate": 5e-5,
    "warmup_steps": 500,
    "max_steps": 10000,
    "batch_size": 16,
    "max_length": 256
}

trainer = Trainer(
    model=model,
    args=TrainingArguments(**model_args),
    train_dataset=train_dataset,
    eval_dataset=val_dataset
)
```

## Example CRAFT Detection Parameters

- Text confidence threshold: 0.7
- Link confidence threshold: 0.4
- Input resolution: 1280x720
- Post-processing NMS threshold: 0.2

## **Appendix D: Additional Evaluation Examples**

This appendix includes additional OCR results to showcase the model's behavior on varied layouts, font types, and Khmer-English mixed inputs.