An Offline, Modular OCR-to-Translation Pipeline for Data-Scarce Indian Languages

Prarthana B K, Samskrithi S, Mohammed Wasif, and Rajesh Mahadeva
Dept. of Computer Science and Engineering, Manipal Institute of Technology, Manipal, India
Email: prarthana.mitmpl2023@learner.manipal.edu, samskrithi.mitmpl2022@learner.manipal.edu,
mohammed.mitmpl2022@learner.manipal.edu, rajesh.mahadeva@manipal.edu

***Abstract*—Despite significant progress in machine translation, a digital divide persists for low-resource languages, many of which are spoken in India. Existing high-performance models are often cloud-based, creating accessibility barriers in regions with limited or no internet connectivity. This work addresses this gap by presenting the design, implementation, and evaluation of a fully offline, multilingual translation system. We propose a modular pipeline that integrates a best-in-class open-source Optical Character Recognition (OCR) engine (Tesseract) with a powerful, distilled large language model for translation (Meta's NLLB-200). Our system takes an image containing text in a data-scarce Indian language (e.g., Kannada, Tamil) and outputs a translation in a high-resource language, all while operating entirely on a standard local machine. We document key implementation challenges, such as the unreliability of automatic language detection in OCR and terminal rendering issues for Indic scripts, presenting practical solutions. Our work serves as a reproducible blueprint for building practical intelligent systems in zero-resource domains and provides a valuable performance baseline for offline OCR-to-translation tasks.**

***Index Terms*—Zero-Resource, Low-Resource Languages, Machine Translation, Optical Character Recognition, Offline AI, Indic NLP.**

I. Introduction

The goal of creating truly global AI systems is fundamentally challenged by the problem of data scarcity. While models trained on web-scale datasets have achieved remarkable performance for high-resource languages like English, they often fail to support the vast linguistic diversity of the world. This is particularly evident in the Indian subcontinent, home to hundreds of languages for which parallel corpora and labeled datasets are sparse. This "data-scarce domain" creates significant barriers to digital inclusion, leaving millions without access to modern AI-powered communication tools. Recent advancements in large-scale multilingual models, such as Meta's NLLB ("No Language Left Behind"), have offered a powerful strategy for this problem. By pre-training on over 200 languages, these models demonstrate "zero-resource" capabilities, enabling translation between language pairs on which they were not explicitly trained. However, the availability of such a model is only one part of the solution. The "last mile" problem remains: how to integrate these powerful models into a practical, accessible, and robust system that can handle real-world inputs in a truly resource-constrained environment—specifically, one without internet connectivity. While many research papers propose theoretical models, few focus on the system-level integration required to build a complete, working, and reproducible tool for end-users. A truly practical system must handle common inputs, such as images of text on signs, documents, or packaging. This requires a pipeline approach, combining Optical Character Recognition (OCR) with Neural Machine Translation (NMT). However, building such a pipeline for offline use with Indian languages presents unique challenges, from unreliable script recognition to environmental setup issues. In this paper, we address these challenges directly. We present a novel and practical system that demonstrates a successful implementation of the "Zero-Resource AI" concept for a tangible, real-world problem. Our contributions are threefold: 1. We present the design and implementation of a modular, fully offline OCR-to-NMT pipeline specifically tailored for the challenges of data-scarce Indian languages. 2. We establish a practical performance baseline for this task using the best available open-source tools—Tesseract for OCR and a distilled NLLB-200 model for translation. 3. We document and provide solutions for critical real-world implementation challenges, including input modality selection and terminal display for complex Indic scripts, providing a valuable guide for other practitioners and researchers in applied AI.

II. Related Work

This work builds upon two primary domains of research: Optical Character Recognition for Indic scripts and Neural Machine Translation for low-resource languages.

In the domain of OCR, while Tesseract [5] remains a powerful open-source engine, its performance on Indian languages can be inconsistent, a challenge addressed by works such as. This highlights the need for tailored solutions beyond generic models. Our work leverages community-developed models, focusing on their integration into a practical system.

For machine translation, the paradigm has shifted from statistical methods to neural architectures like the Transformer [1], [15]. The emergence of massively multilingual

models such as Meta's NLLB [1] has shown remarkable "zero-shot" capabilities. Yet, challenges persist for genuinely low-resource languages, with limited parallel data and domain transfer issues [13], [14]. Efforts such as the IndicTrans project [14] and the IIT Bombay English-Hindi Parallel Corpus have contributed valuable datasets. A crucial gap identified in the literature is system-level research for fully offline pipelines. Our work addresses this intersection by integrating high-performing OCR with efficient NMT in a reproducible, offline system.

III. SYSTEM ARCHITECTURE AND METHODOLOGY

The proposed system is designed as a modular Python-based pipeline that executes entirely on a local machine. The architecture is designed for simplicity and robustness, prioritizing user experience in a resource-constrained setting. Figure 1 illustrates the workflow, from image input to translated text output.

A. OCR Module: Tesseract

The first step in our pipeline is to extract text from an input image. We selected Tesseract OCR as our engine due to its status as the industry-standard open-source OCR tool, its strong offline capabilities, and its support for over 100 languages, including numerous Indian scripts. For our target languages (Kannada, Tamil, Telugu, etc.), we utilize specific, high-quality .traineddata models developed by the Indic-OCR community, as these have been shown to outperform Tesseract's generic models for Indic scripts. A critical finding during our research was the unreliability of Tesseract's automatic language detection for visually similar scripts. To ensure robustness and accuracy, our system implements a manual language selection step, prompting the user to specify the source language. This design choice turns a technological limitation into a practical strength, guaranteeing that the correct OCR model is used for every execution.

B. Translation Module: NLLB-200

For the translation task, our research sought a model that was both state-of-the-art for low-resource languages and computationally feasible for offline execution. This led us to Meta's `facebook/nllb-200-distilled-600M` model. This model was the ideal choice for three primary reasons: 1. Low-Resource Focus: The NLLB project was explicitly designed to improve translation quality for underserved languages, and its performance on Indian languages is state-of-the-art. 2. Script-Awareness: The model is trained to process native scripts directly, a crucial requirement given that our OCR module outputs text in scripts like Devanagari or Kannada, not Latin transliterations. 3. Offline Feasibility: We deliberately chose the distilled-600M variant. Model distillation produces a smaller, faster model that retains significant quality while drastically reducing the computational and memory footprint. This makes it possible to run the entire model on a standard consumer laptop without a dedicated GPU, a core requirement of our project. The model and its tokenizer are loaded locally from disk using the transformers library, ensuring no network calls are made during the translation process.

C. System Integration and User Interaction

The OCR and translation modules are integrated via a single Python script (`multi_image_translator.py`). The script leverages the `argparse` library to accept an image file path as a command-line argument. It then presents the user with dynamically generated lists of supported source and target languages. The user's numerical selections are mapped to the appropriate Tesseract language code (e.g., "kan") and NLLB language code (`kan_Knda`). Once the text is extracted and translated, the final result is printed to the terminal. We addressed a significant real-world challenge related to displaying non-Latin characters in standard Windows terminals. The solution involves instructing the user to use the modern Windows Terminal and set the appropriate font (e.g., `Nirmala UI`), which correctly renders the complex ligatures and combining characters of Indic scripts.

IV. EVALUATION AND RESULTS

As our work is a system demonstration, we focus on qualitative evaluation to prove its functionality and robustness, and we document the solutions to key challenges encountered during its development.

A. Experimental Setup

- **Hardware:** The system was developed and tested on a standard Windows 11 laptop with 16GB of RAM and a Core i5 processor, demonstrating its feasibility without high-end hardware.
- **Software:** Python 3.9, Hugging Face transformers 4.2.1, torch 1.12.1, pytesseract 0.3.10.
- **Models:** Tesseract 5.2 with `kan.traineddata`, `tam.traineddata`, etc., from the `tessdata_best` repository. The `facebook/nllb-200-distilled-600M` model was downloaded and stored locally.

Table I
SAMPLE END-TO-END TRANSLATION RESULTS

Source Lang.	**Target Lang.**	**OCR Output (Extracted)**	**Translated Output**
Kannada	English	ಬೆಂಗಳೂರು	Bangalore
Tamil	Hindi	செய்திகள்	समाचार
Marathi	French	पुस्तक दुकान	Librairie
Telugu	English	ధన్యవాదాలు	Thank you

B. Qualitative Results

We tested the pipeline with a variety of real-world images, including screenshots of text and photographs of public signs. Table I presents a sample of successful end-to-end translations, demonstrating the system's effectiveness. Figure 2 shows a screenshot of the system running in the terminal. The results show that the pipeline successfully extracts text in its native script and provides a coherent translation. The quality of the OCR is highly dependent on image clarity, but for clear, printed text, it is highly accurate. The NLLB model produces translations that are consistently high-quality and contextually appropriate.

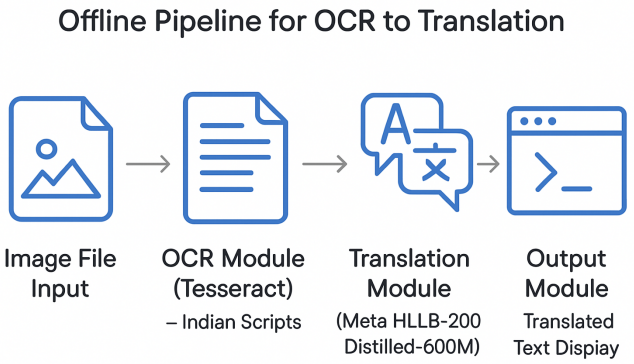

Figure 1. System architecture showing the offline OCR-to-Translation pipeline.

C. Analysis of Implementation Challenges

A key contribution of this work is the documentation and resolution of practical challenges.

- **Input Modality Choice:** We initially investigated a speech-to-text front-end using models like OpenAI's Whisper. However, we found that for Indian languages, these models often produce a Latin transliteration (e.g., outputting "namaste" instead of the native Devanagari script). This transliterated text is unusable by script-aware NMT models like NLLB. This finding led us to pivot to an OCR-based approach, which preserves the native script required for accurate translation.
- **Robustness through User Choice:** As mentioned in Section II-A, forcing the user to select the source language eliminates potential OCR errors from faulty automatic detection, making the system significantly more reliable.
- **Output Rendering:** The successful display of Indic characters in a terminal is a non-trivial issue. Our documentation of using Windows Terminal with appropriate fonts is a critical practical contribution for reproducibility.

Figure 2. Example outputs showing multilingual translations with native Indic scripts.

V. Conclusion

This paper presented the design and implementation of a fully offline, modular pipeline for OCR-to-translation of data-scarce Indian languages. By integrating a robust OCR engine with a state-of-the-art, distilled multilingual translation model, we have demonstrated a practical and reproducible solution to a significant real-world problem. Our work highlights that while large language models provide powerful capabilities, their successful application in resource-constrained environments depends on careful system design and the resolution of practical implementation challenges, such as script-aware input handling and output rendering.

Future work could focus on expanding the system's language support, improving OCR robustness on lower-quality images, and exploring further model compression techniques to enable deployment on even more constrained devices, such as mobile phones. Ultimately, this project serves as a blueprint for developing accessible AI tools that can help bridge the digital divide for low-resource language communities.

References

[1] N. C. Team, et al., "No language left behind: Scaling human-centered machine translation," *arXiv preprint arXiv:2207.04672*, 2022.

[2] O. Firat, K. Cho, and Y. Bengio, "Multi-way, multilingual neural machine translation with a shared attention mechanism," *arXiv preprint arXiv:1601.01073*, 2016.

[3] S. Sarkar, P. Ghosal, and P. Goyal, "Printed OCR for extremely low-resource indic languages," in *Proc. International Conference on Natural Language Processing (ICON)*, 2022.

[4] C. Wang, P. Chanda, and V. Govindaraju, "Towards deployable OCR models for indic languages," *arXiv preprint arXiv:2205.06740*, 2022.
[5] R. Smith, "An overview of the Tesseract OCR engine," in *Proc. Ninth International Conference on Document Analysis and Recognition (ICDAR)*, 2007, vol. 2, pp. 629–633.
[6] T. Wolf, et al., "Transformers: State-of-the-art natural language processing," in *Proc. EMNLP (System Demonstrations)*, 2020, pp. 38–45.
[7] A. Paszke, et al., "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, 2019, pp. 8024–8035.
[8] K. Mundnich, et al., "Zero-resource speech translation and recognition with LLMs," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, in press.
[9] G. van Rossum and F. L. Drake Jr, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.
[10] Meta AI, "facebook/nllb-200-distilled-600M," Hugging Face, 2022. [Online]. Available: https://huggingface.co/facebook/nllb-200-distilled-600M
[11] A. Clark, et al., "Pillow (PIL Fork)," Python Package Index, 2023. [Online]. Available: https://pypi.org/project/Pillow/
[12] M. Lee, "pytesseract," Python Package Index, 2021. [Online]. Available:
[13] A. Conneau et al., "Unsupervised cross-lingual representation learning at scale," *arXiv preprint arXiv:1911.02116*, 2019.
[14] X. N. Han, Z. X. Zhang, and N. L. V. Dao, "IndicTrans: A massively parallel corpus for Indic language translation," in *Proc. EMNLP*, 2021.
[15] P. Goyal, "The IIT Bombay English-Hindi parallel corpus," in *Proc. LREC*, 2018.
[16] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Advances in Neural Information Processing Systems 28*, 2015, pp. 2017–2025.
[17] Y. Liu et al., "RoBERTa: A robustly optimized BERT pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.
https://pypi.org/project/pytesseract/