



International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Design of Sanskrit Devanagari Script Recognition and Sanskrit to English Translation System using deep learning.

Dr. Aniruddha Kailuke¹, Rutuja Deshmukh², Bhavesh Bargat³, Shital Tijare⁴, Akshad Kurve⁵

Department of Artificial Intelligence & Data Science Engineering, Priyadarshini College of Engineering, Nagpur (R.T.M. Nagpur University), Maharashtra, India

ABSTRACT :

Sanskrit literature contains valuable knowledge in the fields of philosophy, linguistics, science, mathematics, medicine, and history. However, many Sanskrit documents are still preserved only as printed books, scanned pages, or handwritten manuscripts, making them difficult to search, edit, and translate. This review paper studies deep learning methods for designing a Sanskrit Devanagari script recognition and Sanskrit-to-English translation system. The objective of the system is to accept a Devanagari document image, recognize the Sanskrit text in editable Unicode form, and translate it into meaningful English prose. The reviewed methodology follows a modular pipeline consisting of image preprocessing, OCR recognition, text post-processing, tokenization, and neural machine translation. OpenCV-based preprocessing techniques such as grayscale conversion, Gaussian blur, Otsu thresholding, and morphological operations improve document clarity. For OCR, CNN/ResNet feature extraction, Bi-LSTM sequence modeling, and CTC decoding are suitable because they avoid character-level segmentation problems caused by shirorekha, matras, and conjunct characters. For translation, Transformer-based models such as IndicBART, mBART, or mT5 can be fine-tuned using Sanskrit-English parallel corpora and SentencePiece tokenization to handle Sandhi, Vibhakti, and flexible word order. The review finds that a two-stage OCR-plus-NMT architecture is practical, maintainable, and suitable for project implementation. Future work should focus on larger Sanskrit OCR datasets, improved manuscript restoration, grammar-aware tokenization, and human-in-the-loop correction.

Keywords Sanskrit OCR; Devanagari script recognition; CNN-BiLSTM-CTC; IndicBART; neural machine translation; SentencePiece; deep learning.

INTRODUCTION

Sanskrit is recognized as the structural bedrock of classical Indian literature, historical records, mathematical treatises, and philosophical discourses. For millennia, it served as the primary vehicle for India's scientific and cultural heritage. However, millions of historical manuscripts remain locked away in physical archives, libraries, and museums, rendering them completely inaccessible to a global modern audience due to a profound linguistic and digital barrier. The overarching goal of this project is to bridge this gap by designing a unified, automated software platform powered by deep learning. The proposed system creates a structural bridge between physical ancient texts and contemporary readers. The architecture relies on an explicit distinction between the visual writing script and the underlying grammatical language: * Devanagari (The

Script): A purely visual writing system consisting of explicit structural symbols, characters, vowels, and consonants. Recognizing it is strictly a Computer Vision (CV) problem of pixel-level classification and sequence decoding. * Sanskrit (The Language): A highly complex linguistic system governing vocabulary, semantic meaning, syntax, and intricate grammatical rules. Translating it into English is strictly a Natural Language Processing (NLP) sequence-to-sequence problem. By treating these tasks as separate modules, the system architecture avoids an unmanageable monolithic neural network, opting instead for a highly maintainable, modular assembly-line pipeline.

TAXONOMY OF VISUAL AND LINGUISTIC CHALLENGES

Generic OCR engines and general-purpose translation systems show a sharp drop in precision when applied to Sanskrit due to unique visual and linguistic constraints.

A. Vision-Level OCR Challenges

Traditional character scanners isolate pixels by searching for clear vertical gaps between individual characters. This assumption breaks down in Devanagari due to the script's connected and layered structure.

1. **Shirorekha interconnection:** Devanagari characters are bound by a continuous top horizontal line called the shirorekha. It fuses letters into a word block and makes standard vertical segmentation unreliable.
2. **Complex conjunct clusters:** Sanskrit uses multi-consonant clusters such as क्ष, त्र, ज्ञ, and श्र. These forms often merge geometrically into ligatures

that cannot be separated by simple character-splitting logic.

3. **Overlapping vowel modifiers:** Matras attach above, below, before, or after base characters and may overlap with the shirorekha or spill into adjacent character regions.

B. Linguistic-Level NMT Challenges

Even when text is recognized accurately, translation into English remains difficult because Sanskrit is morphologically rich and syntactically flexible.

1. **Sandhi compounding rules:** Words undergo phonetic and morphological merging at boundaries. A continuous printed string can contain multiple hidden words.
2. **Morphological case inflection:** Nouns, pronouns, and verbs change endings according to case, gender, number, tense, and grammatical role, increasing vocabulary density.
3. **Flexible word order:** Case endings define the role of words, allowing flexible placement in a sentence. This confuses purely left-to-right phrase translation models.

SYSTEM DEVELOPMENT METHODOLOGIES

Current automated transcription and translation frameworks use a decoupled sequential pipeline where the OCR stage passes recognized Unicode text to the NMT stage through a strict digital interface.

A. Image Preprocessing and Matrix Normalization

Raw image payloads are ingested into memory arrays where OpenCV filters eliminate color noise channels. RGB data is flattened into a single 8-bit grayscale channel using the luminance equation: $\text{Gray} = 0.299R + 0.587G + 0.114B$. High-frequency scanning noise is reduced using Gaussian filtering. Instead of using a fixed threshold, Otsu's thresholding analyzes the pixel histogram and automatically determines a binarization limit to separate ink from background noise. Morphological dilation and erosion are then applied to connect broken strokes and detach touching characters.

B. Spatial Feature Mapping and Segment less OCR Core

The binary image is divided into horizontal text rows using projection profiling or contour-based line extraction. Each row is passed into a deep CNN architecture such as ResNet-18. The network scans the line image and converts pixel grids into deep visual feature maps. These feature maps are reshaped into sequential vertical feature vectors along the width of the line.

The feature sequence is routed into a Bidirectional Long Short-Term Memory network. The Bi-LSTM evaluates text context from left to right and right to left, which helps recognize overlapping character clusters and matras. The output is decoded by a Connectionist Temporal Classification layer. CTC avoids explicit character bounding boxes and maps temporal feature sequences directly into UTF-8 Devanagari Unicode strings.

C. Sub-Word Tokenization and Transformer Machine Translation

The raw OCR text is normalized before being processed by a Sentence Piece sub-word tokenizer. Word-level tokenizers are insufficient for Sanskrit because Sandhi and compounds blend separate words into large surface strings. Sentence Piece breaks these strings into reusable sub-word units.

The tokenized vectors are passed into a sequence-to-sequence Transformer encoder-decoder model. Multi-Head Self-Attention captures grammatical relationships between words regardless of position, making the architecture suitable for Sanskrit's free word order. The decoder uses cross-attention to generate fluent English sentences and is optimized using cross-entropy loss with AdamW.

COMPREHENSIVE DATA AND HARDWARE TOPOLOGIES

A. Open-Source Labeled Datasets

4. **OCR track dataset:** The Devanagari Handwritten Character Dataset contains approximately 92,000 images across 46 Devanagari consonant and numeral classes and can serve as a baseline for visual character training.
5. **Translation track dataset:** The Itihasa Sanskrit-English parallel corpus provides more than 93,000 aligned Sanskrit shloka and English translation pairs from the Ramayana and Mahabharata.

B. Hardware and Compute Infrastructure

Large Transformer configurations and deep convolutional networks often require GPU acceleration. Development may use Google Colab Pro, Kaggle notebooks, NVIDIA T4 or A100 GPUs, CUDA, and cuDNN. Code collaboration and version management can be maintained through GitHub. Model deployment can be wrapped using FastAPI and exposed through a Streamlit dashboard.

REVIEW MATRIX SUMMARY

ID	Academic Source	Primary Architecture	Key Project Takeaway
1	IEEE Xplore / recent deep learning OCR-NMT studies	Hybrid CNN-RNN + Transformer	Dual sequential OCR and translation pipelines are viable for end-to-end Sanskrit processing.
2	Devanagari segmentation studies	Median profile + ResNet/CNN	Shirorekha handling improves character separation and line tracking.
3	Transformer-based Sanskrit translation studies	Attention Transformer	Transformers handle non-linear and long-range Sanskrit dependencies better than older recurrent approaches.
4	IEEE Devanagari OCR survey	Manual features + SVM/HMM	Handwriting variation requires robust post-processing and error-tolerant correction.
5	Handwritten character benchmark studies	HOG + SVM	Classical descriptors are useful baselines but are weaker than deep convolutional features.
6	ACL IndicBART	Indic sequence-to-sequence Transformer	Orthographic similarity across Indic languages improves transfer learning for low-resource generation.

METHODOLOGICAL EVALUATION METRICS

Scientific evaluation of the system requires separate metrics for the OCR and translation modules.

1. Character Error Rate (CER): Evaluates character-by-character OCR errors using $CER = (Schar + Dchar + Ichar) / Nchar$, where S, D, and I represent substitutions, deletions, and insertions.
2. Word Error Rate (WER): Tracks word-level sequence deformation using $WER = (Sword + Dword + Iword) / Nword$.
3. BLEU Score: Evaluates final English translation quality using n-gram precision with brevity penalty: $BLEU = BP \times \exp(\sum w_n \log p_n)$.

For Sanskrit translation, human evaluation is also important because multiple valid English renderings may exist for a single Sanskrit sentence.

OPEN CHALLENGES AND FUTURE DIRECTIONS

A. Transition to Vision Transformers

Current CRNN frameworks rely on localized CNN line slicing. Vision Transformers can divide pages into patches and track global context, which may improve recognition of curved, distorted, or irregular manuscript lines.

B. Generative Reconstruction Models

Historical palm-leaf and paper manuscripts may contain missing ink, physical tears, stains, and faded strokes. Generative Adversarial Networks or diffusion-based restoration models could reconstruct missing strokes before OCR classification.

C. Grammar-Aware Translation

Sanskrit translation can improve through Sandhi splitting, morphological tagging, and grammar-aware tokenization. Combining neural translation with Sanskrit grammatical analyzers may reduce semantic errors.

D. Human-in-the-Loop Correction

Allowing users to manually correct OCR output before translation can reduce error propagation and improve final English output quality.

CONCLUSION

This survey reviewed deep learning paradigms for Sanskrit Devanagari OCR and Sanskrit-to-English translation. The analysis shows that a modular pipeline is the most practical architecture: image preprocessing and CNN-BiLSTM-CTC recognition generate Unicode Devanagari text, and a

Transformer-based NMT system translates it into English. The key challenges are shirorekha connectivity, conjunct clusters, matras, Sandhi, Vibhakti, and free word order. Future research should emphasize line-level Sanskrit OCR datasets, Vision Transformers, manuscript restoration, grammar-aware preprocessing, and human-in-the-loop correction.

REFERENCES

1. X. Zhang, Y. Wang, and L. Mishra, "Design of a Sanskrit Devanagari Script Recognition and Sanskrit Translation System Based on Deep Learning," IEEE Xplore Digital Library, vol. 24, no. 2, pp. 114–121, Feb. 2025.
2. R. Sharma, K. Purohit, and S. Joshi, "A Systematic Framework for Sanskrit Character Recognition Using Deep Learning," Electronic Letters on Computer Vision and Image Analysis (ELCVIA), vol. 24, no. 1, pp. 81–94, Jan. 2025.
3. A. Kumar and P. Dave, "Sanskrit to English Translation: A Comprehensive Survey and Implementation using Transformer Based Model," International Journal of Computer Applications, vol. 185, no. 28, pp. 14–22, May 2024.
4. R. Jayadevan, S. R. Kolhe, P. M. Patil, and U. Pal, "Online Recognition of Devanagari Script: A Survey," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 41, no. 6, pp. 782–796, Nov. 2011.
5. S. Dhankhar, M. K. Gupta, F. H. Memon, S. Bhatia, P. Dadheech, and A. Mashat, "Support Vector Machine Based Handwritten Hindi Character Recognition and Summarization," Computer Systems Science and Engineering, vol. 43, no. 1, pp. 397–412, May 2022.
6. D. Kakwani et al., "IndicBART: A Pre-trained Model for Indic Natural Language Generation," in Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 2022, pp. 3125–3136.