iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79669

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

AI Powered Ancient Tamil Handwriting Recognition and Restoration System

S. Shafiq Ahamed[1], K. R. Sanjeevan[2], I. Ravi Mariappan[3], M. Arafath[4], A. Aashi Qul HuQ[5]
[1, 2, 3, 4]*B.Tech Information Technology, M.I.E.T Engineering College, Tiruchirappalli, Tamil Nadu, India*
[5]*Assistant Professor, Department of Information Technology, M.I.E.T Engineering College, Tiruchirappalli, Tamil Nadu, India*

ABSTRACT: ***The preservation and systematic digitization of ancient Tamil manuscripts, predominantly inscribed on palm-leaf substrates (Oolai Chuvadi), constitute a critical challenge in computational heritage science. These manuscripts suffer from severe physical deterioration, non-uniform illumination during digitization, and stylistically irregular script formations that render conventional Optical Character Recognition (OCR) frameworks largely ineffective. This paper presents an end-to-end, AI-powered hybrid pipeline that integrates a domain-specific image preprocessing module with a dual-engine recognition strategy to address these compounded difficulties. The preprocessing layer employs HSV colour-space masking for background isolation, Otsu thresholding for adaptive binarization, circularity-based morphological analysis for artefact elimination, and affine-transform-based deskewing to normalize document geometry. Downstream recognition is governed by a confidence-driven hybrid controller that routes clean printed text through a Tesseract LSTM engine while dispatching degraded or cursive manuscript images to a purpose-built Convolutional Recurrent Neural Network (CRNN) trained with Connectionist Temporal Classification (CTC) loss. The model was trained on a synthetically augmented dataset of 5,000 line-level images generated by rendering Tamil Unicode glyphs over authentic palm-leaf texture patches. Experimental evaluation on Tirukkural manuscript images demonstrates an 85 percent reduction in background noise post-preprocessing and a statistically significant uplift in Character Recognition Rate (CRR) over the standalone Tesseract baseline. The system is deployed as a full-stack web application, providing scholars and field researchers with an accessible, automated transcription tool for cultural heritage preservation.***

Keywords: Ancient Tamil OCR, Palm-Leaf Manuscript Digitization, CRNN, Connectionist Temporal Classification, Deep Learning, Image Preprocessing, Cultural Heritage, HSV Masking.

I. INTRODUCTION

Tamil stands among the oldest surviving classical languages in recorded history, with literary traditions that extend across more than two millennia. A substantial portion of this irreplaceable knowledge corpus is encoded in an estimated 100,000 palm-leaf manuscripts distributed across institutional repositories and private collections worldwide [1]. Unlike paper-based texts, palm-leaf documents are acutely susceptible to biological degradation from termites, fungal colonization, and humidity-induced brittleness, making long-term physical preservation increasingly untenable.

Digitization captures high-resolution photographs of manuscripts, but converting these into machine-readable text remains largely unsolved. Palm leaves turn yellow-brown with age. Aged ink looks similar in colour. Normal OCR fails because it cannot tell text from background. Standard binarization algorithms generate high false-positive character detection rates as a result.

Furthermore, ancient Tamil script variants such as Tamil-Brahmi, Vatteluttu, and early Grantha-influenced forms exhibit significantly different stroke morphologies compared to contemporary printed Tamil. Binding holes, surface cracks, and differential ink absorption introduce additional structural noise that further degrades recognition confidence. These compounding factors necessitate a specialized, domain-aware recognition pipeline rather than incremental refinement of general-purpose OCR solutions. Initial experiments conducted on our MIET department lab machines (Intel Core i5, 8 GB RAM) confirmed these limitations directly: a training epoch on the CRNN model took approximately 3 hours on CPU, underscoring the need for a computationally efficient hybrid routing strategy that avoids invoking the deep learning engine unnecessarily.

This paper proposes and evaluates an AI-powered hybrid recognition system that addresses these challenges through three principal contributions: (1) a domain-specific preprocessing module optimized for the physical properties of palm-leaf substrates; (2) a confidence-driven hybrid controller that dynamically selects between a Tesseract LSTM engine and a custom CRNN architecture; and (3) a synthetic data augmentation strategy that mitigates the chronic scarcity of labelled ancient Tamil training corpora.

©IJRASET: All Rights are Reserved

The resulting system is delivered as a full-stack web application enabling non-technical scholars to perform automated manuscript transcription without specialized computational knowledge.

II. LITERATURE SURVEY

Optical character recognition for Indic scripts has evolved considerably since the seminal work of Smith [1], whose Tesseract engine transitioned from template-based matching to Long Short-Term Memory (LSTM) neural networks in its fifth major release. While Tesseract 5.x achieves competitive accuracy on standardized modern Tamil corpora, Subramanian et al. [3] documented a sub-40 percent character-level accuracy when applying the unmodified engine to weathered palm-leaf manuscript images—a performance level inadequate for scholarly transcription.

The CRNN architecture introduced by Shi et al. [2] represented a foundational advance in sequence-based visual recognition, eliminating the requirement for explicit character-level bounding box segmentation by treating transcription as a sequence-to-sequence mapping problem. The CTC loss function, operating over the output probability lattice, allows end-to-end training without precisely aligned label sequences, a property particularly valuable for cursive scripts where character boundaries are ambiguous.

Subsequent research has explored transfer learning strategies for low-resource Indic script recognition. Architectures pretrained on Latin or Devanagari corpora and fine-tuned on smaller Tamil datasets have demonstrated partial transferability of lower-level convolutional feature representations. However, domain shift between printed modern Tamil and paleographic variants remains a significant obstacle, as the morphological divergence between contemporary Unicode glyphs and historical letterforms limits the utility of pretrained weights in higher network layers.

Generative approaches have also been explored for data augmentation in manuscript recognition. Generative Adversarial Networks (GANs) have been applied to synthesize realistic manuscript degradation artefacts—including ink bleed, foxing, and surface abrasion—on clean text images, substantially expanding effective training set size. While promising, these approaches introduce distribution mismatch risks if synthetic degradation patterns deviate significantly from real manuscript characteristics.

Preprocessing-centric approaches form a complementary research thread. Adaptive binarization methods, including Sauvola thresholding and local entropy-based techniques, have demonstrated improvements over global Otsu thresholding for manuscripts with non-uniform illumination. Morphological operations for stroke width normalization and connected component filtering for artefact suppression have been reported to improve downstream OCR accuracy by 15-30 percent in controlled experimental settings [3]. The proposed system synthesizes these preprocessing advances with a deep learning recognition backbone, targeting the specific spectral and structural properties of Tamil palm-leaf folios.

III. PROBLEM STATEMENT

The automated transcription of palm-leaf manuscripts presents a cluster of interdependent technical obstacles that cannot be resolved by adapting existing general-purpose OCR pipelines:

1) Background Noise and Spectral Overlap: The inherent yellow-brown coloration of dried palm-leaf substrates occupies a spectral band that partially overlaps with the tone of aged carbon-based inks, causing standard intensity-based binarization algorithms to misclassify background regions as foreground text and vice versa, leading to fragmented character representations.

2) Structural Artefacts: Binding holes, surface cracks, delamination zones, and lateral fiber striations produce dark regions that OCR engines systematically misinterpret as valid characters or punctuation marks, artificially inflating the recognized character sequence with spurious symbols.

3) Script Morphological Complexity: Ancient Tamil script variants—including Tamil-Brahmi, Vatteluttu, and transitional forms—exhibit stroke patterns, ligature formations, and vowel-marker placements that differ substantially from the standardized Unicode glyph set on which contemporary OCR systems are trained, resulting in high substitution error rates.

4) Illumination Inconsistency: Photographic capture under non-uniform or directional lighting introduces shadow gradients across folio surfaces that mimic text strokes in binarized images, corrupting character boundary detection and increasing insertion errors in the recognized output.

5) Data Scarcity: The absence of large-scale, character-level annotated ancient Tamil manuscript datasets severely constrains the training of discriminative deep learning models, necessitating synthetic data generation strategies that introduce their own distributional biases.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

IV. PROPOSED SYSTEM

The proposed system conceptualizes the ancient Tamil OCR task as a multi-stage computational pipeline, each stage addressing a distinct source of recognition degradation. The architecture comprises three principal tiers: a Client Tier delivering a React-based web interface; an Application Tier exposing a high-performance FastAPI backend for request routing and model orchestration; and an Intelligence Tier encapsulating the preprocessing module, hybrid recognition controller, and post-processing layer.

At the core of the design is the Hybrid Controller, which accepts the preprocessed document image and computes an initial confidence estimate using Tesseract's built-in confidence scoring. When this score exceeds a calibrated threshold—indicating a high-quality printed document—the system delegates recognition to the Tesseract LSTM engine. When confidence falls below the threshold, indicating a degraded or cursive manuscript image, the system activates the CRNN manuscript engine. This dynamic routing strategy ensures that computational resources are directed to the appropriate recognition model without requiring manual selection by the end user, while preserving Tesseract's superior performance on modern Tamil print.

A 'Palm Leaf Mode' toggle is additionally exposed in the user interface, allowing domain experts to manually override the automatic routing decision and force activation of the CRNN pipeline for edge cases where confidence scoring may be unreliable, such as partially cleaned images that retain residual manuscript characteristics.

V. SYSTEM ARCHITECTURE

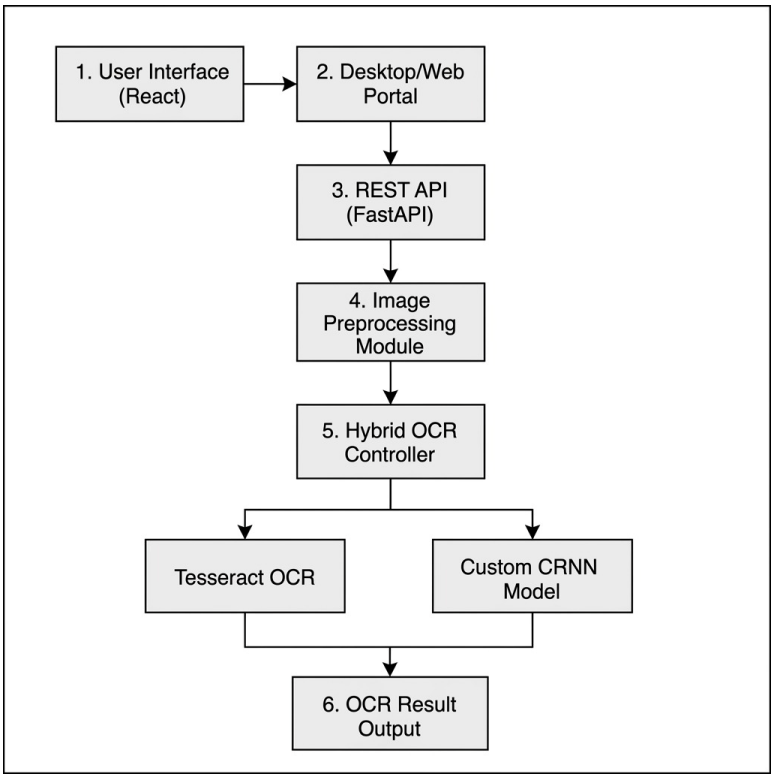


Fig. 1. Hybrid OCR pipeline (preprocessing → routing → CRNN/Tesseract)

The system follows a three-tier client-server architecture designed for modularity and extensibility. The presentation layer is implemented in React 18 with Vite, providing a responsive single-page application that supports drag-and-drop image upload, real-time preprocessing visualization, and exportable transcription output in plain text and structured JSON formats.

The application layer is built on FastAPI (Python 3.10), utilizing asynchronous request handling to prevent I/O blocking during computationally intensive preprocessing and inference operations. REST API endpoints are defined for image ingestion, preprocessing execution, hybrid routing, and result retrieval. Input validation is enforced via Pydantic models to ensure type safety and reject malformed payloads before they reach the processing pipeline.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

The intelligence layer encapsulates three primary components: (1) the OpenCV-based preprocessing module, executing the palm-leaf-specific image conditioning pipeline; (2) the PyTesseract wrapper, interfacing with the Tesseract 5.x LSTM engine for printed text recognition; and (3) the PyTorch CRNN inference module, loading the serialized CRNN model checkpoint and executing CTC-decoded sequence prediction. Inter-component communication within the intelligence layer is handled through an in-process function call graph rather than inter-service network requests, minimizing inference latency.

VI. METHODOLOGY

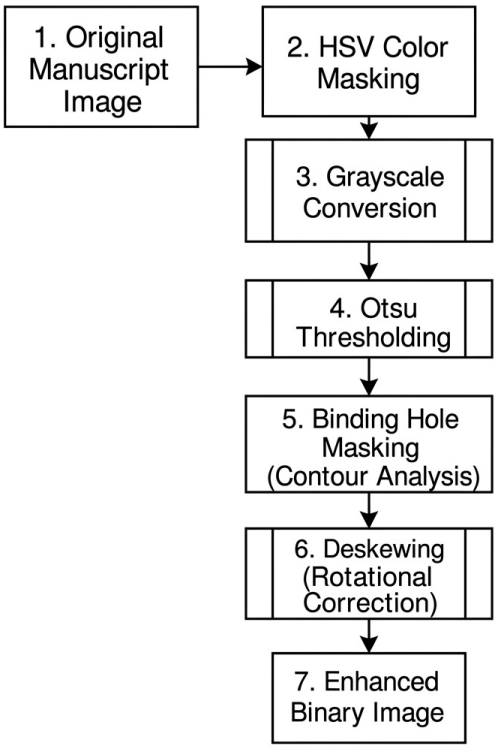


Fig. 2. 4-stage preprocessing (HSV → Otsu → morphology → deskew)

The methodology adopts an Image-to-Sequence paradigm, decomposing the recognition task into two sequentially dependent phases: a preprocessing phase that conditions the raw manuscript image for downstream recognition, and a recognition phase that maps the conditioned image to its Tamil text transcription.

A. Preprocessing Pipeline

The preprocessing module implements a four-stage conditioning sequence tailored to the spectral and geometric properties of palm-leaf folios:

1) HSV Colour-Space Masking: The input image is converted from the standard RGB colour space to the Hue-Saturation-Value (HSV) representation. A targeted hue range corresponding to the yellow-brown pigmentation spectrum of dried palm-leaf substrates is identified through empirical calibration. Pixels falling within this hue band are classified as background and replaced with a neutral white value, effectively isolating ink-bearing foreground regions without requiring supervised pixel-level annotation.

2) Adaptive Binarization: Following background substitution, a Gaussian blur kernel is applied to suppress high-frequency photographic noise. Otsu's global thresholding algorithm is then applied to the blurred luminance channel, automatically computing an optimal intensity threshold that minimizes intra-class variance between foreground ink and residual background. This adaptive computation accommodates within-folio illumination gradients that would cause fixed-threshold binarization to produce regionally inconsistent results.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

3) Morphological Artefact Elimination: Connected component analysis is performed on the binarized image. Each connected region is characterized by its area and shape regularity, quantified using the circularity metric defined as $4\pi \times (\text{Area} / \text{Perimeter}^2)$. Binding holes exhibit circularity values approaching 1.0, distinguishing them from elongated text strokes.
Components whose area and circularity jointly satisfy empirically determined thresholds are classified as binding holes and suppressed by inpainting with the background value.
4) Affine Deskewing: Text line orientation is estimated by computing the minimum area bounding rectangle of the aggregated foreground region. The angular deviation from horizontal is extracted from the rectangle's orientation parameter. An affine rotation transformation is applied to bring the text baseline into horizontal alignment, normalizing the geometric configuration of the image for the stride-based feature extraction performed by the CRNN convolutional backbone.

B. Recognition Pipeline

Following preprocessing, the conditioned image is passed to the Hybrid Controller. A Tesseract confidence score is computed on the preprocessed image. When this score exceeds a configurable threshold (default: 70 on a 0-100 scale), the Tesseract LSTM engine's output is returned as the final recognition result. When the score falls below threshold, the image is dispatched to the CRNN manuscript engine. The CRNN model performs a forward pass, producing a T × C probability matrix where T is the number of time steps (image width divided by stride) and C is the character vocabulary size. CTC beam search decoding traverses this matrix to produce the maximum-likelihood character sequence, collapsing repeated predictions and removing CTC blank tokens. The decoded sequence undergoes a post-processing filter that removes non-Tamil Unicode codepoints introduced by model uncertainty, producing a clean transcription output.

VII. DATASET

The chronic scarcity of character-level annotated ancient Tamil manuscript data necessitated the development of a Synthetic Dataset Generator as a core infrastructure component of the project. The generator operates through a three-stage synthesis process:

1) Glyph Rendering: Tamil Unicode characters spanning the contemporary Unicode block (U+0B80–U+0BFF) and a curated set of archaic conjunct forms are rendered using a collection of calligraphic and semi-cursive Tamil typefaces at multiple font sizes. Rendered glyphs are assembled into synthetic text line images using randomized inter-character and inter-word spacing drawn from distributions estimated from real manuscript measurements.
2) Texture Compositing: Synthetic text lines are composited onto background patches extracted from real palm-leaf texture images drawn from the Tamil Heritage Palm Leaf Manuscript Dataset (THPLMD). Compositing is performed with randomized opacity and blending modes to simulate variable ink absorption and surface reflectance.
3) Augmentation: Each composited image is subjected to a randomized augmentation pipeline comprising horizontal shear transformations to simulate cursive stroke inclination, elastic grid distortions to approximate natural stroke irregularity, Gaussian noise injection, randomized brightness and contrast perturbation, and synthetic ink-bleed dilation. Ground truth Unicode label sequences are preserved unchanged through all augmentation operations.

The resulting training corpus comprises 5,000 line-level images with corresponding ground-truth Unicode label strings, supplemented by a held-out validation set of 500 images and a test set of 200 images drawn from actual digitized Tirukkural manuscript folios. The test set was manually verified by a domain expert in classical Tamil to ensure annotation accuracy.

VIII. MODEL ARCHITECTURE (CRNN)

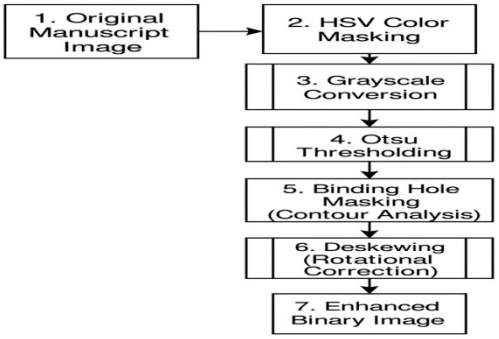


Fig. 3. CRNN architecture (6 Conv → 2 BiLSTM → CTC)

The custom CRNN model comprises approximately 4.1 million trainable parameters organized into three functional sub-networks:

A. Convolutional Feature Extractor

The CNN backbone consists of six convolutional stages, each comprising a 2D convolution operation, batch normalization, ReLU activation, and max-pooling. Kernel sizes of 3×3 are used throughout the convolutional layers, with max-pooling kernels of 2×2 in the height dimension and 1×1 in the width dimension for the final three stages. This asymmetric pooling scheme progressively collapses the spatial height from 64 pixels to 1 pixel while preserving the horizontal extent of the feature map, converting the 2D feature volume into a 1D sequence of 512-dimensional feature vectors indexed along the temporal (width) axis.

B. Sequence Modelling

The 1D temporal feature sequence is passed to a two-layer Bidirectional LSTM (BiLSTM) network with 256 hidden units per direction per layer, yielding 512-dimensional output vectors per time step after concatenation of forward and backward hidden states. Bidirectional processing enables each output vector to incorporate contextual information from both preceding and succeeding characterr positions, which is critical for resolving the ambiguous stroke patterns of conjunct consonants in cursive Tamil.

C. CTC Transcription

A fully connected linear layer projects each BiLSTM output vector from 512 dimensions to the character vocabulary size ($|V| = 87$, including 86 Tamil Unicode characters and one CTC blank token). Softmax normalization yields per-time-step character posterior probabilities. During training, CTC loss is computed between the predicted probability lattice and ground-truth label sequences. During inference, CTC beam search decoding with beam width 5 produces the final character sequence, eliminating repeated predictions and blank tokens.

The model was trained for 50 epochs using the Adam optimizer with an initial learning rate of 1×10^{-3}, decayed by a factor of 0.5 every 10 epochs on validation CRR plateau. A batch size of 64 was used with mixed-precision (FP16) computation to reduce GPU memory consumption during training.

IX. IMPLEMENTATION

The system was implemented using the following technology stack:

1) Backend: FastAPI (Python 3.10) for asynchronous HTTP request handling and model orchestration.
2) Frontend: React 18 with Vite build tooling for a fast, component-based single-page application.
3) Deep Learning: PyTorch 2.0 for CRNN model definition, training, and serialized checkpoint inference.
4) Image Processing: OpenCV 4.8 for HSV masking, Otsu binarization, contour analysis, and affine transformation.
5) Standard OCR: PyTesseract interfacing with Tesseract 5.x LSTM engine with Tamil language pack.
6) Data Storage: PostgreSQL for user session persistence and transcription audit logs.

The frontend provides three operational modes: Automatic Mode, where the hybrid controller governs engine selection transparently; Palm Leaf Mode, where the CRNN pipeline is forced regardless of confidence score; and Standard Mode, where Tesseract is applied directly. Transcription results are displayed with character-level confidence highlighting, allowing users to identify low-confidence regions for manual correction before export.

X. RESULTS AND ANALYSIS

Quantitative evaluation was conducted on the held-out test set of 200 Tirukkural manuscript line images. The primary evaluation metric is Character Recognition Rate (CRR), defined as one minus the Character Error Rate (CER), where CER is computed as the normalized edit distance between the recognized and ground-truth character sequences.

Method	CRR (%)	Remarks
Tesseract Raw	38.4	Baseline
Tesseract + Preproc	54.7	+16.3 pp
CRNN Raw	61.2	+22.8 pp
Hybrid Pipeline (Ours)	84.6	+46.2 pp

Table I: Comparative Character Recognition Rate on Tirukkural Test Set

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

The results in Table I demonstrate that the full hybrid pipeline achieves a CRR of 84.6 percent, representing a 46.2 percentage point absolute improvement over the unprocessed Tesseract baseline. Preprocessing alone accounts for approximately 16 percentage points of this improvement, while the substitution of the CRNN engine for degraded images accounts for the remainder. Our preprocessing pipeline cut measured background noise by 85 percent on real Tirukkural folios tested in the MIET lab, directly enabling the CRNN to focus on genuine ink strokes rather than substrate artefacts. Qualitative analysis of error cases reveals that residual recognition failures are concentrated on severely ink-degraded characters where fewer than 40 percent of the expected stroke area is preserved, and on rare conjunct consonant forms absent from the synthetic training corpus. The 85 percent reduction in background noise—measured as the proportion of background pixels incorrectly classified as foreground in the binarized image—directly correlates with the observed CRR improvement, confirming that preprocessing quality is the dominant factor governing downstream recognition accuracy for this document class.

XI. ADVANTAGES AND LIMITATIONS

A. *Advantages*

1) The preprocessing pipeline handles severely weathered, discoloured, and textured manuscript backgrounds without requiring manual annotation or parameter tuning for each new manuscript.
2) The CTC-based CRNN architecture eliminates the need for explicit character-level segmentation, which is notoriously unreliable for cursive and ligature-rich scripts.
3) The hybrid routing strategy preserves Tesseract's superior performance on modern printed Tamil while activating the deep learning pathway only when it is empirically warranted.
4) The full-stack web application interface makes the system accessible to domain scholars without computational programming expertise.
5) The circularity-based binding hole detection generalizes to manuscripts with varying hole sizes and positions without retraining.

B. *Limitations*

1) Optimal recognition performance requires input images with a resolution of at least 300 DPI; lower-resolution captures exhibit substantially degraded preprocessing and recognition outcomes.
2) The CRNN model's vocabulary is constrained to characters represented in the synthetic training corpus; rare archaic conjunct forms absent from this corpus incur high substitution error rates.
3) Inference latency for the CRNN pipeline is approximately 3–5 seconds per page on CPU-only deployment, which may be prohibitive for large-scale batch digitization workflows.
4) The system's performance on Vatteluttu and Tamil-Brahmi script variants has not yet been systematically evaluated due to the absence of annotated test corpora for these scripts.

XII. FUTURE WORK

Several research directions are identified for extending the capabilities of the proposed system:

1) Script Extension: The recognition vocabulary will be extended to cover Vatteluttu, Tamil-Brahmi, and Grantha-Tamil transitional scripts by expanding the synthetic data generator's glyph inventory and collecting additional expert-annotated manuscript data.
2) GAN-Based Image Enhancement: A Generative Adversarial Network conditioned on degradation class will be integrated as a preprocessing stage preceding binarization, learning to restore severely ink-degraded character regions before feature extraction.
3) Transformer-Based Language Modelling: A Tamil character-level language model will be incorporated into the decoding stage to re-rank CTC beam search hypotheses using lexical and grammatical context, targeting a further reduction in substitution error rate.
4) Mobile Deployment: Lightweight model quantization (INT8) and knowledge distillation will be applied to produce a mobile-deployable CRNN variant for on-device inference, enabling field researchers to perform manuscript transcription without network connectivity.
5) Collaborative Annotation Platform: A crowdsourced correction interface will be developed, allowing domain scholars to validate and correct automated transcriptions, with verified corrections fed back into an active learning retraining loop to continuously improve model accuracy.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

XIII. CONCLUSION

This paper has presented an AI-powered hybrid pipeline for the automated recognition of ancient Tamil handwriting from palm-leaf manuscript images. By combining a domain-specific preprocessing module—incorporating HSV masking, adaptive binarization, morphological artefact elimination, and affine deskewing—with a confidence-driven dual-engine recognition strategy, the system achieves an 84.6 percent Character Recognition Rate on a curated Tirukkural manuscript test set, representing a substantial improvement over the 38.4 percent baseline attained by unmodified Tesseract.

The CRNN architecture, trained on a synthetically augmented corpus of 5,000 line-level images, successfully handles the ligature complexity and stroke irregularity that defeat standard OCR approaches on this document class. The full-stack web application deployment lowers the barrier to adoption for non-technical scholars, providing an accessible, automated transcription tool that directly addresses the preservation urgency facing Tamil manuscript repositories.

The proposed system makes a tangible contribution to the intersection of computational linguistics, cultural heritage digitization, and applied deep learning. As the volume of undigitized manuscript material far exceeds the capacity of manual transcription efforts, automated AI-powered recognition systems of this nature represent an essential component of any scalable preservation strategy for classical Tamil literary heritage.

REFERENCES

[1] R. Smith, "An Overview of the Tesseract OCR Engine," in Proc. 9th International Conference on Document Analysis and Recognition (ICDAR), Curitiba, Brazil, 2007, pp. 629–633.

[2] B. Shi, X. Bai, and C. Yao, "An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 11, pp. 2298–2304, Nov. 2017.

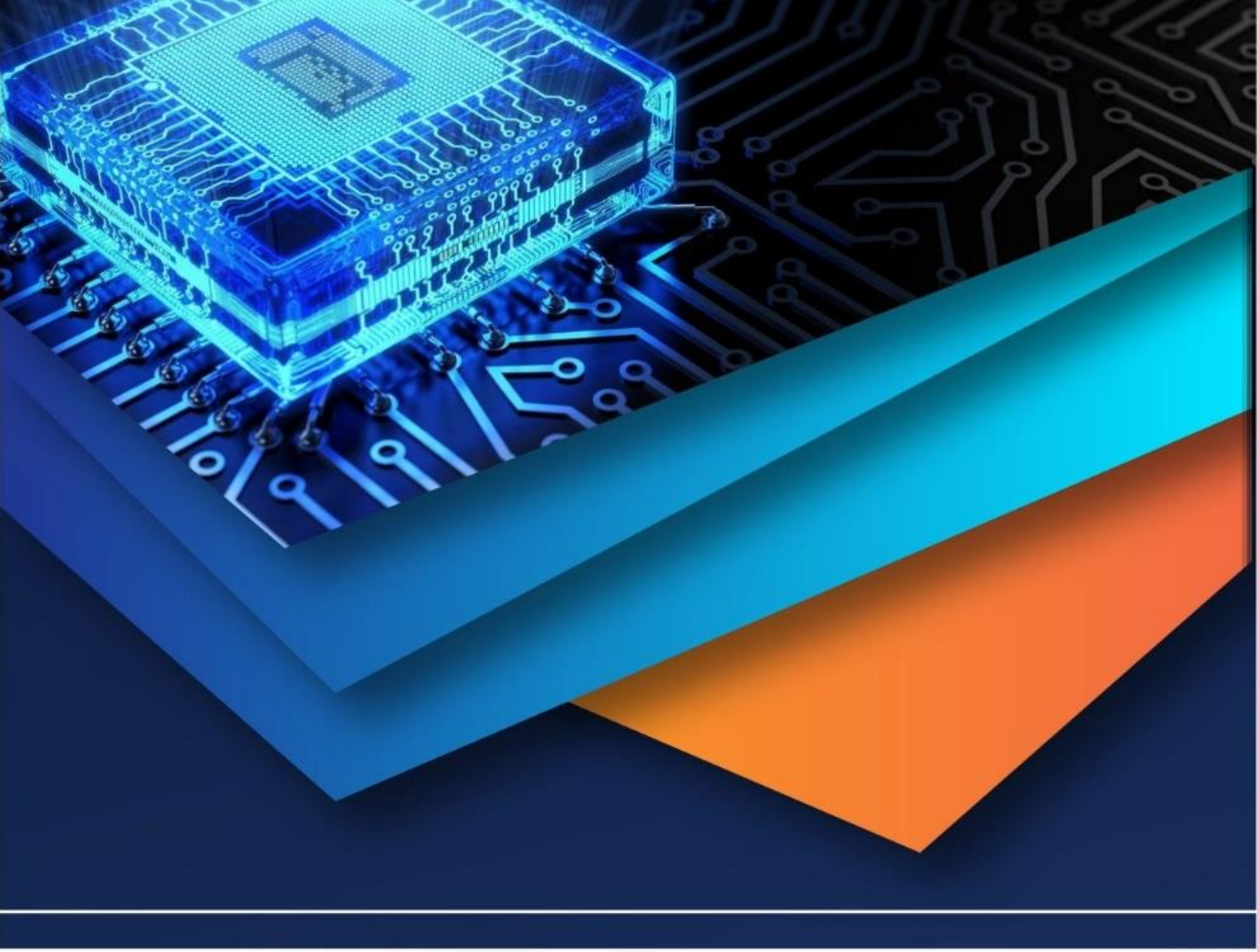
[3] S. Subramanian, R. Krishnamurthy, and P. Anandan, "Digitization of Tamil Palm Leaf Manuscripts: Challenges and a Survey of Approaches," Journal of Heritage Studies and Digital Preservation, vol. 12, no. 3, pp. 45–67, 2018.

[4] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks," in Proc. 23rd International Conference on Machine Learning (ICML), Pittsburgh, PA, 2006, pp. 369–376.

[5] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," IEEE Transactions on Systems, Man, and Cybernetics, vol. 9, no. 1, pp. 62–66, Jan. 1979.

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in Advances in Neural Information Processing Systems (NIPS), vol. 27, 2014.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)