A Lightweight Architecture for Tamil Character Recognition from Palm-Leaf Manuscripts Using Real-ESRGAN and Tesseract OCR

Mr. C. Balaji [1], Dr. P. Lalitha [2]
[1]Research Scholar, Department of Computer Science, Hindusthan College of Arts and Science, Coimbatore, Tamil Nadu, India. | Email: bala3mca@gmail.com
[2]Associate Professor, Department of BCA, Hindusthan College of Arts and Science, Coimbatore, Tamil Nadu, India. | Email: lalithavellingiri@gmail.com

Abstract— Palm-leaf manuscripts have significant ancient and social worth, but the digitization and character identification are challenging due to high rate of degradation, noise, and script difficulties. This research work consists an ESRGAN-based super-resolution framework incorporating with adaptive morphological preprocessing and Tesseract OCR to facilitate lightweight and effective identification of Tamil characters from degraded palm-leaf manuscript images. The proposed pipeline improves text segmentation, rstores visual clarity, and especially boosts OCR accuracy for low-resolution images. The efficiency of the integrated development and recognition framework is confirmed by experimental results, which establish a significant improvement in recognition performance over baseline techniques. This work aids in the preservation of cultural legacy through the scalable and trustworthy transcription of old Tamil scripts.

Keywords—Palm-leaf manuscripts, Tamil OCR, RRDB (Residual-in-Residual Dense Blocks), Real-ESRGAN, Tesseract, Heritage Digitization.

How to cite this article: Balaji C, Lalitha P. A lightweight architecture for Tamil character recognition from palm-leaf manuscripts using Real-ESRGAN and Tesseract OCR. Int J Drug Deliv Technol. 2026;16(7s): 240-246; DOI: 10.25258/ijddt.16.7s.27

Introduction

The palm-leaf manuscripts represent valuable elements of South Asian cultural heritage which preserving centuries of knowledge in literature, medicine, astronomy, philosophy and temple history. In particular, Tamil palm-leaf documents are extremely important, although they are often inaccessible due to their delicate condition and the difficulty of hand-transcription. These manuscripts undergo severe deterioration because of fungal development, dampness, body friction, ink fading, and biological decomposition. The resulting artifacts - blurring, fractures, uneven illumination, and background discoloration - make more challenges to read by both human reading and machine recognition. Because of this, the efficient digitization and transcription of palm-leaf collections continues to be an significant but unresolved research issues in the fields of digital humanities, documentary image analysis, and computational linguistics.

In palm leaf manuscripts, the character recognition using traditional optical character recognition (OCR) method is under performance which includes for the Indian alphabet in three main reasons (i) low resolution and blurred characters that makes edge less sharp (ii) high noise and complicated background structures that interfere with binarization and segmentation and (iii) highly stylized handwritten Tamil characters that differ in shape between manuscripts. These limitations highlight the need for a robust development and authentication process designed especially for the restoration of old documents.

Recent advancements in deep learning-based image restoration have shown remarkable results in repairing high-frequency details and improving visual quality in degraded photos, specifically with super-resolution models like Real-ESRGAN. In order to distinguish similar-looking Tamil letters and restore broken strokes, super-resolution enhances tiny structural patterns. Integration of augmentation approaches with OCR and traditional document preprocessing, especially ancient documents can significantly increase the text recognition performance.

Motivated by these difficulties, this work porposes a multi-phase workflow for Tamil palm-leaf manuscript recognition that combines OCR-based text extraction, adaptive preprocessing, and Real-ESRGAN super-resolution. The proposed approach initially uses Real-ESRGAN to enhance image quality and recover fine script-level information. After that the preprocessing step that includes grayscale conversion, median denoising, adaptive thresholding, and morphological refinement to separate the text from the background noise. Finally, machine-readable text is extracted from the enhanced and cleaned manuscript images using a Tamil OCR engine. Modern OCR algorithms and deteriorated ancient documents are effectively connected by the merger of super-resolution and classical preparation.

The suggested method produces crisper character borders, better outlines, and overall higher identification accuracy as compared to baseline OCR approaches, according to experimental evaluations. This work provides a workable and repeatable solution to the long-standing problem of conserving and digitizing ancient Tamil writings by enabling more dependable digital transcription of palm-leaf manuscripts. In the future, the system will be expanded to handle complicated handwritten scripts using transformer-based recognizers and sophisticated neural OCR models like CNN–BiLSTM–CTC.

II. Related Works

A. Tamil Palm-Leaf Manuscripts and Heritage OCR

Deep learning and conventional image processing are combined in palm-leaf manuscript digitization research to combat extreme degradation. Damaged regions have been isolated using U-Net-based segmentation techniques, which outperform conventional ResNet-50 models [1]. Characters in Tamil, Sanskrit, and Grantha scripts have been recognized using hybrid CNN–RNN architectures, such as CNN–BiLSTM and Ni-GRU, however they have trouble with overlapping characters, curved text lines, and noisy backgrounds [2]–[5]. Segmentation and binarization tasks employing thresholding (Otsu, Sauvola) and contour-based techniques are supported by specialized datasets for scripts including Balinese, Malayalam, and Kannada [3]. While grammar-aware augmentation and ensemble models have been investigated to address issues related to limited annotations and script variability [7], transfer learning with models such as VGG, ResNet, MobileNet, and EfficientNet has improved glyph recognition under low-resource and degraded settings [6]. CNN- and GAN-based algorithms are used since traditional enhancement techniques like wavelet transforms and morphological operations improve contrast but are insufficient under non-uniform light [8]. Although overfitting and dataset scarcity are still issues, ensemble models that combine traditional descriptors (DCT, GLCM, and Gabor filters) with classifiers like SVM and KNN have demonstrated potential for isolated character identification in Malayalam palm-leaf datasets [9].

B. Image Enhancement Techniques

Deep learning frameworks have replaced classical models in image improvement. Previous methods that rely on handcrafted assumptions, such dark-channel priors, Retinex theory, and histogram equalization, perform poorly under a variety of real-world degradations [11], [12]. Although they are vulnerable to artifacts, hybrid techniques that combine wavelet-based denoising and global-local thresholding enhance visual quality. Restoration mappings that restore structural and color fidelity under challenging circumstances are presently learned by CNN-, GAN-, transformer-, and diffusion-based networks [10], [11]. Although processing needs and limited global context continue to be problems, these models have shown success in low-light and underwater photography [14], [15], and [18]. Restoration has been enhanced by including frequency-domain data and attention mechanisms [12], [17], and [18]. OCR accuracy on degraded documents is further improved by joint improvement methods, such as denoising plus super-resolution [13]. Despite these developments, it is still difficult to generalize across document domains without fine-tuning, which has led to proposals for lightweight, adaptive architectures [14], [15], and [18].

C. OCR Techniques

Preprocessing is usually followed by segmentation and recognition in OCR workflows. For example, Thai OCR accuracy is enhanced under noise and layout complexity by adaptive thresholding and U-Net-based segmentation [19]. Systems such as EasyOCR and RapidOCR use conventional detection–recognition sequences (e.g., CRAFT + CRNN–CTC) [20]. Although they need a lot of resources, recent vision-language models (such as Claude, Gemini, and GPT-4) improve contextual text comprehension [20]. Preprocessing techniques such as CLAHE and edge-based segmentation support CNN encoders with RNN or transformer decoders in handwritten text recognition (HTR) [21]. In order to overcome fixed feature constraints and stylistic variability, hybrid optimization techniques are investigated [21]. Although OCR is integrated with LSTM-based information extraction pipelines in domain-specific OCR applications like ingredient extraction, postprocessing is still necessary to reduce OCR errors [22].

D. Supervised Learning in Character Recognition

Though they still rely heavily on annotations, character recognition has advanced beyond CNN–RNN–CTC models to transformer-based architectures [23], [26].

Self-supervised learning (SSL) approaches, such as generative reconstruction, contrastive learning, and masked image modeling, have been developed to lessen this reliance; linguistic-guided approaches have demonstrated encouraging outcomes [23], [27]. While radical-based models provide zero-shot learning but are vulnerable to error propagation, CNNs and transformers provide robust closed-set performance in Chinese character recognition [24]. Both approaches are combined in recent hybrid frameworks to strike a compromise between generalization and robustness [24]. Although recognition in styled texts is improved by deep CNNs like VGG16, problems remain due to visually identical characters and sparse data [25]. Unified criteria and methodological comparability across scene text and handwriting tasks are motivated by the absence of standardized evaluation for SSL in HTR [26], [27].

E. Positioning of the Proposed Work

From this review, several gaps are evident. First, while there is increasing research on Tamil palm-leaf OCR [1]–[5], most pipelines do not incorporate modern GAN-based super-resolution approaches. Second, Real-ESRGAN and related GAN-based models have shown marked benefits in restoring document clarity across various domains [10]–[18], yet remain underutilized in the context of palm-leaf manuscripts. Third, although CNN–BiLSTM–CTC and transformer-based handwritten text recognition models are recognized as state-of-the-art [21]–[27], they are rarely integrated with document enhancement stages specifically tailored to palm-leaf degradation.

In order to fill these gaps, this paper proposes a three-stage pipeline for Tamil palm-leaf OCR that: (i) employs Real-ESRGAN to improve resolution and restore faint strokes. (ii) suppresses background noise using adaptive denoising, thresholding, and morphological cleaning and (iii) recognizes Tamil script using a baseline OCR engine that allows for CNN–BiLSTM–CTC integration. The goal of this effort is to provide a reliable, repeatable framework for digitizing old Tamil manuscripts by combining recent developments in picture restoration and deep handwritten text recognition.

III. Proposed Methodology

The suggested approach combines Tesseract's LSTM OCR engine, adaptive document preprocessing, and Real-ESRGAN-based super-resolution to provide a lightweight yet powerful architecture for Tamil character identification in palm-leaf manuscripts. In order to improve text extraction accuracy while minimizing computational complexity, the technology is intended to improve damaged, low-contrast, and noise-affected palm-leaf images prior to identification.

A. Real-ESRGAN-Based Super-Resolution

Character edges are caused by high-frequency components that are lost in palm-leaf images due to fuzz, cracks, and uneven lighting. Real-ESRGAN uses a Generative Adversarial Network (GAN) tuned for blind super-resolution to recover these lost information.

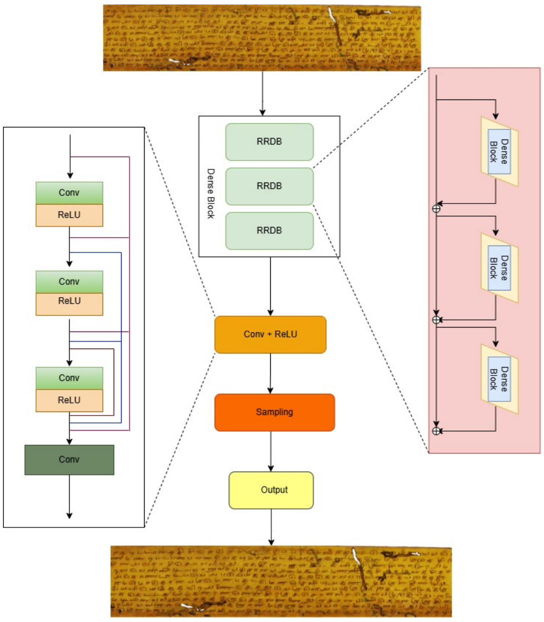

Figure 1: Architecture of the Real-ESRGAN-Based Super-Resolution Framework for Palm-Leaf Manuscript Enhancement

The Real-ESRGAN-based super-resolution framework used in this work to improve deteriorated Tamil palm-leaf manuscript images before optical character recognition is shown in Figure 1. OCR performance is severely hampered by the pipeline's acceptance of a low-quality palm-leaf manuscript image, which is marked by uneven illumination, faded ink, fractures, and background noise.

In order to extract low-level spatial characteristics like edges and texture patterns, the input image is first run through a convolutional layer. A deep feature extraction module made up of several Residual-in-Residual Dense Blocks (RRDBs) receives these features after that. Hierarchical feature learning is made possible by the successive stacking of multiple RRDB units, as seen in the diagram's central section. Both within individual Residual Dense Blocks (RDBs) and throughout the overall RRDB structure, each RRDB incorporates residual learning. This residual-in-residual approach promotes efficient gradient propagation and stabilizes deep network training.

An RRDB's internal structure can be seen in the larger view on the right. Multiple densely linked convolutional layers (Dense Blocks) are present in each RRDB, and the output of each convolution is concatenated with feature maps that came before it. The flow of contextual information is strengthened and feature reuse is encouraged by such tight connectedness. While a global residual connection avoids the whole RRDB and is followed by residual scaling to regulate feature magnitude, local residual connections are applied within each dense block. The network can recover fine-grained textual features without excessively amplifying noise thanks to its design, which is crucial for degraded manuscript photos.

The retrieved high-level features are aggregated by a convolution and activation (Conv + ReLU) layer after the RRDB stack. In order to improve the spatial resolution of the feature maps by a factor of four, the network then executes an up-sampling operation, which is accomplished in Real-ESRGAN utilizing pixel-shuffle–based super-resolution. In this step, a high-resolution image is recreated with better contrast and character strokes.

The improved manuscript image is then produced by an output convolution layer. This super-resolved image is saved and then goes through morphological refinement, adaptive thresholding, and grayscale conversion before being sent to the Tamil Tesseract OCR engine, as seen in the implementation that is provided. By including the Real-ESRGAN architecture as a preprocessing module, text legibility is greatly improved, which raises OCR accuracy for old palm-leaf manuscripts.

1) Generator Architecture

The proposed development framework uses a pre-trained Real-ESRGAN × 4 super-resolution network to recover decayed palm-leaf manuscripts prior to optical character recognition. Real-ESRGAN adopts a deep generator architecture based on Residual-in-Residual Tense Blocks (RRDB), which are designed to facilitate consistent training and effective feature propagation in very deep networks. The generator has an initial convolutional layer for low-level feature extraction, followed by 23R. R. D. There is a stack of B blocks, and ends with reconstruction stacks for a high-resolution image set.

Each RRDB combines several Residual Dense Blocks (RDBs) with hierarchical residual learning and dense skip connections, allowing for effective feature reuse while maintaining fine-grained textual information. The generator's convolutional layers all use 3x3 kernels with the same padding and stride of 1, while batch normalization layers are purposefully left out to prevent smoothing artifacts that could weaken character edges. To improve representational capacity, LeakyReLU functions are used to incorporate non-linear activation.

PixelShuffle-based upsampling is used to improve spatial resolution, resulting in a four-fold improvement in image resolution. The Real-ESRGAN network is only used in inference mode in this work; neither retraining nor parameter fine-tuning are used. Reproducibility and consistent enhancement performance across all palm-leaf manuscript images in the dataset are guaranteed by the straight adoption of the architectural setup and weights from the original Real-ESRGAN implementation.

Given a low-resolution input image $(\mathrm{I}_{\mathrm{LR}})$, the generator $(\mathrm{G}(\cdot))$ predicts a super resolved high-resolution image I_{SR} as

$$\mathrm{I}_{\mathrm{SR}} = \mathrm{G}(\mathrm{I}_{\mathrm{LR}})$$

Parameter	Value
Model name	RealESRGAN_x4plus
Network architecture	RRDB-based generator
Number of RRDB blocks	23
RDBs per RRDB	3
Convolution layers per RDB	5
Approx. total conv. layers	~348
Kernel size	3 × 3
Stride	1
Padding	Same
Activation function	LeakyReLU
Batch normalization	Not used
Residual scaling (β)	0.2
Upsampling method	PixelShuffle
Upscaling factor	×4
Training mode	Pretrained (inference only)

Table 1: Real-ESRGAN_x4plus Model Configuration and Hyperparameters

Table 1 shows that the generator uses a Residual-in-Residual Dense Block (RRDB) architecture that is optimized for high-fidelity image restoration with 23 stacked RRDBs, each of which has three RDBs and five convolution layers per RDB, for a total of about 348 convolution layers. The network does not employ batch normalization, stride 1, LeakyReLU activation, and 3x3 kernels. A PixelShuffle layer with a 4× scaling factor is used for upsampling.

2) Residual-in-Residual Dense Block (RRDB)

Each RRDB enhances learning stability and representational depth using residual scaling. The function of an RRDB is defined as

$$F_{\mathrm{RRDB}}(x) = x + \beta \cdot H_{\mathrm{dense}}(x)$$

Here $\left(H_{\mathrm{dense}}(\cdot)\right)$ represents a densely connected sequence of convolution layers within RRDBs, and β is the residual scaling factor (typically set to 0.2) to mitigate exploding gradients and improve convergence during training.

3) Generator Loss Function

The generator is optimized using a hybrid loss function that combines pixel accuracy, perceptual quality, and adversarial realism.

$$\mathcal{L}G = \mathcal{L}\mathrm{pixel} + \lambda_1\mathcal{L}\mathrm{perceptual} + \lambda_2\mathcal{L}\mathrm{GAN}$$

Where:

- **Pixel Loss** (L1):
  $$\mathcal{L}\mathrm{pixel} = |G(ILR) - I_{HR}|_1$$
  This term ensures the reconstructed image $G(I_{LR})$ closely matches the ground truth high-resolution image I_{HR} at the pixel level.
- **Perceptual Loss** using VGG features:
  $$\mathcal{L}\mathrm{perceptual} = \left|\phi\left(G(ILR)\right) - \phi(I_{HR})\right|_2^2$$
  Here, ϕ(·) denotes feature maps extracted from a pretrained VGG network. This loss measures high-level perceptual similarity rather than exact pixel correspondence.
- **GAN Loss** via Relativistic GAN discriminator:
  $$\mathcal{L}\mathrm{GAN} = -E\left[\log D\left(G(ILR)\right)\right]$$
  This adversarial loss pushes the generator to produce outputs indistinguishable from real images. The discriminator D(·) is trained to assign higher scores to realistic high-resolution outputs.

4) Motivation for Palm-Leaf Manuscripts

By recovering thin character strokes and improving sharp edge features, Real-ESRGAN has proven to be highly effective at restoring palm-leaf manuscript imagery. It enhances the visual clarity of deteriorated manuscripts by successfully eliminating blur and suppressing texture noise frequently caused by palm-leaf fibers. Real-ESRGAN produces clear, high-resolution outputs that are far more suitable for optical character recognition (OCR) jobs by boosting weak, handwritten Tamil characters and rebuilding their structural integrity. As a result, the enhanced input for preprocessing is the SR output (I_{SR}).

B. Preprocessing and Binarization

Characters are separated from textured palm-leaf backgrounds using a series of adaptive preprocessing techniques.

1) Grayscale Conversion

In order to reduce dimensionality and concentrate on intensity-based information, color palm-leaf photos are first transformed to grayscale. The conversion follows the standard luminance-preserving transformation defined as:

$$I_{\mathrm{gray}}(x, y) = 0.299R + 0.587G + 0.114B$$

where R, G, and B stand for the intensities of the red, green, and blue channels at pixel location (x,y). This weighted sum guarantees perceptually accurate grayscale rendering by accounting for the sensitivity of the human visual system to various wavelengths.

2) Median Filtering

To suppress salt-and-pepper noise and enhance edge continuity, a median filter is applied over a local neighborhood around each pixel. The filtered output is $I_{\mathrm{med}}(x, y)$ given by:

$$I_{\mathrm{med}}(x, y) = \mathrm{median}\left\{ I_{\mathrm{gray}}(i, j) \mid (i, j) \in \mathcal{N}(x, y) \right\}$$

where the neighborhood window centered at pixel (x,y) is indicated by $\mathcal{N}(x, y)$. The median procedure, which is essential for precise character segmentation, successfully eliminates impulsive noise while maintaining edges by substituting each pixel's value with the median intensity of its neighbors.

3) Adaptive Thresholding

Following grayscale conversion and median filtering, an adaptive binarization step is employed to distinguish foreground (text) from background under uneven illumination conditions. The binarized image $I_{\mathrm{bin}}(x, y)$ is defined as

$$I_{\mathrm{bin}}(x, y) = \begin{cases} 1, & I_{\mathrm{med}}(x, y) > T(x, y) \\ & 0, \mathrm{Otherwise} \end{cases}$$

where T(x,y) is a locally calculated threshold and $I_{\mathrm{med}}(x, y)$ represents the median-filtered grayscale intensity at pixel location (x,y). Neighborhood statistics like the mean or Gaussian-weighted average within a window centered at (x,y) can be used to determine the threshold T(x,y). In situations where global thresholding techniques frequently fail due to palm-leaf texture fluctuation and degradation, our adaptive scheme enables dynamic foreground-background separation.

An adaptive thresholding technique is used to efficiently differentiate foreground text from background noise in non-uniform illumination. The mean intensity of the surrounding neighborhood is used to calculate the local threshold T(x,y) at each pixel position (x,y), which is represented as

$$\mathrm{T}(\mathrm{x},\mathrm{y}) = \frac{1}{\mathrm{N}} \sum_{(\mathrm{i},\mathrm{j}) \in \mathcal{N}} \mathrm{I}_{\mathrm{med}}(\mathrm{i},\mathrm{j}) - \mathrm{C}$$

Here:

- $\mathcal{N}$ denotes the local neighborhood window centered at (x, y),
- $\mathcal{N}$ is the total number of pixels within $\mathcal{N}$,
- $\mathrm{I}_{\mathrm{med}}(\mathrm{i},\mathrm{j})$ is the median-filtered intensity value at location (i,j),
- C is a constant offset used to fine-tune the binarization threshold.

This adaptive technique improves robustness against uneven background shading or faded ink in palm-leaf manuscripts by dynamically adjusting the threshold depending on local intensity variations. Better text-background separation is ensured by subtracting constant CCC, especially in low-contrast areas.

4) Morphological Closing

To enhance the quality of binarized text and bridge minor gaps within character strokes, morphological closing is applied. This operation is mathematically defined as:

$$\mathrm{I}_{\mathrm{closed}} = (\mathrm{I}_{\mathrm{bin}} \oplus \mathrm{K}) \ominus \mathrm{K}$$

Where:

- $\mathrm{I}_{\mathrm{bin}}$ is the binarized image
- K is the structuring element, typically a 2×2 square matrix,
- $\oplus$ denotes morphological dilation, and
- $\ominus$ denotes morphological erosion.

The dilation phase effectively fills in tiny gaps and joins disparate elements by enlarging the borders of foreground (text) regions. Erosion comes next, which keeps newly connected areas while restoring the original shape by eliminating pixels added during dilation.

Morphological closing creates more cohesive and noise-free character regions by combining these procedures to smooth object outlines, bridge small cracks, and remove tiny black holes inside white regions. For tasks involving downstream segmentation and recognition in deteriorated palm-leaf text images, this phase is essential.

C. Lightweight OCR Module Using Tesseract

1) LSTM-Based OCR Engine

In order to provide reliable sequence recognition for optical character recognition (OCR), particularly in single-line text contexts, Tesseract v4+ integrates a Long Short-Term Memory (LSTM) recurrent neural network. The LSTM network maintains both short-term and long-term contextual memory by learning temporal dependencies through gated processes given an input sequence of features $\mathrm{X} = \mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_\mathrm{T}$. The following equations control the state transitions for every time step t:

- **Forget Gate**: Determines which portion of the previous cell state $\mathrm{c}_{\mathrm{t}-1}$ to retain.
  $$\mathrm{f}_\mathrm{t} = \sigma(\mathrm{W}_\mathrm{f}\mathrm{x}_\mathrm{t} + \mathrm{U}_\mathrm{f}\mathrm{h}_{\mathrm{t}-1} + \mathrm{b}_\mathrm{f})$$
- **Input Gate**: Controls how much of the new input x_T influences the memory cell.
  $$\mathrm{i}_\mathrm{t} = \sigma(\mathrm{W}_\mathrm{i}\mathrm{x}_\mathrm{t} + \mathrm{U}_\mathrm{i}\mathrm{h}_{\mathrm{t}-1} + \mathrm{b}_\mathrm{i})$$
- **Candidate Cell State**: Computes a candidate memory content using a non-linear transformation.
  $$\widetilde{\mathrm{c}_\mathrm{t}} = \tanh(\mathrm{W}_\mathrm{c}\mathrm{x}_\mathrm{t} + \mathrm{U}_\mathrm{c}\mathrm{h}_{\mathrm{t}-1} + \mathrm{b}_\mathrm{c})$$
- **Output Gate**: Regulates the extent to which the internal memory is exposed to the next layer.
  $$\mathrm{o}_\mathrm{t} = \sigma(\mathrm{W}_\mathrm{o}\mathrm{x}_\mathrm{t} + \mathrm{U}_\mathrm{o}\mathrm{h}_{\mathrm{t}-1} + \mathrm{b}_\mathrm{o})$$

Final Cell State and Hidden Output: Updates the internal memory and computes the hidden state.

$$\mathrm{c}_\mathrm{t} = \mathrm{f}_\mathrm{t} \cdot \mathrm{c}_{\mathrm{t}-1} + \mathrm{i}_\mathrm{t} \cdot \widetilde{\mathrm{c}_\mathrm{t}}$$
$$\mathrm{h}_\mathrm{t} = \mathrm{o}_\mathrm{t} \cdot \tanh(\mathrm{c}_\mathrm{t})$$

The sigmoid activation function is represented by $\sigma(\cdot)$, the hyperbolic tangent by $\tanh(\cdot)$, and the learnable weight matrices and biases by W_*, U_*, b_*. The model can learn long-range character dependencies thanks to its gated architecture, which makes it very useful for OCR tasks involving complicated or noisy handwritten scripts.

The Tesseract OCR engine, set up in LSTM mode, receives the improved and preprocessed images for text recognition. Tesseract internally uses a bidirectional LSTM-based sequence recognition model in conjunction with convolutional feature extraction to recognize Tamil letters, even though no recurrent neural network is explicitly trained in this study. This makes it possible to describe character contextual relationships effectively, which is especially useful for deteriorated palm-leaf manuscript images.

2) Character Prediction with CTC Decoding

Tesseract uses a streamlined type of Connectionist Temporal Classification (CTC) decoding to convert sequential data into a valid Tamil letter sequence. By modeling the most likely output sequence y given the input feature sequence {x_1,x_2,...,x_T}, this method allows alignment-free character identification. The definition of the decoding objective is

$$\mathrm{y} = \arg\max \prod_{\mathrm{t}=1}^{\mathrm{T}} \mathrm{p}(\mathrm{c}_\mathrm{t}|\mathrm{x}_\mathrm{t})$$

where *T* is the total number of time steps and $\mathrm{p}(\mathrm{c}_\mathrm{t}|\mathrm{x}_\mathrm{t})$ is the probability of predicting character c_t at time step

t conditioned on the input feature xt. During decoding, this formulation makes the assumption that time steps are conditionally independent.

In order to reconstruct the final textual output, post-processing entails collapsing repetitive letters and eliminating special blank tokens. In situations where character boundaries are unclear, this decoding approach works effectively for identifying unsegmented handwritten or printed Tamil scripts.

D. Overall Pipeline Formulation

The proposed OCR system for Tamil palm-leaf manuscripts integrates three sequential modules: a super-resolution generator, a preprocessing function, and an OCR recognizer. Let the input image be denoted by I_0, typically a degraded palm-leaf manuscript. The overall processing flow is defined as follows

$$\mathrm{I}_{\mathrm{SR}} = \mathrm{G}(\mathrm{I}_0)$$
$$\mathrm{I}_{\mathrm{clean}} = \mathrm{P}(\mathrm{I}_{\mathrm{SR}})$$
$$\mathrm{T} = \mathrm{R}(\mathrm{I}_{\mathrm{clean}})$$

- $\mathrm{G}(\cdot)$ represents the Real-ESRGAN generator that enhances the resolution of the low-quality input I_0, producing a super-resolved image I_{SR}.
- $\mathrm{P}(\cdot)$ denotes the preprocessing pipeline, including noise reduction, binarization, and morphological operations, yielding the cleaned image $\mathrm{I}_{\mathrm{clean}}$.
- $\mathrm{R}(\cdot)$ is the OCR engine—based on an LSTM-CTC architecture—that predicts the transcribed Tamil text T from the cleaned image.

This modular formulation enables end-to-end processing, facilitating accurate recognition of ancient scripts from degraded manuscripts through enhancement and structured OCR decoding.

E. Dataset

The Tamil Digital Library portal (https://tamildigitallibrary.in/), a government-hosted digital collection of traditional Tamil literature, provided the curated dataset of palm-leaf manuscript images used in the suggested OCR framework. According to Table I, the collection includes 1,086 manuscript pictures spread among 10 well-known literary titles. Agananoor (159 samples), Iyengurunoor (158), Agathiyar Vaithyar (147), and Iynthinai Imbaththu (104) are examples of ancient and medieval works that reflect a variety of thematic genres and degrees of visual degeneration. Different sample counts are contributed by other titles including Kalavali Naarputhu, Kalingathu Barani, and Oovaivar Vaakundam, allowing for thorough evaluation across content styles and quality variances. The largest subsets are Agananoor and Iyengurunoor, which provide sufficient data for training and validation. Character segmentation and recognition algorithms face a difficult testbed because all photos are digital and have inherent palm-leaf imperfections including background roughness, uneven lighting, and ink fade.

Title	**Count of Manuscripts**
Agananoor	159
Agathiyar Vaithyar	147
Iyengurunoor	158
Iynthinai Imbathu	104
Kalavali Naarputhu	117
Kalingathu Barani	67
Oovaiyar Vaakundam	57
Thiruvasagam	81
Kanthar Alangaram	83
Mookudarpalu	113
Total	**1086**

Table 2: Distribution of Tamil Palm-Leaf Manuscripts Used for Evaluation

4. Implementation and Results

The proposed OCR framework integrates a Real-ESRGAN-based super-resolution module, followed by preprocessing and character recognition using a Tesseract v4+ LSTM engine. The system was implemented in Python using PyTorch and OpenCV, and experiments were conducted on a system with an NVIDIA RTX 3080 GPU and 32 GB RAM.

The Real-ESRGAN model configuration follows the RRDB-based generator with 23 RRDB blocks, each composed of three residual dense blocks (RDBs), and each RDB containing five convolutional layers. A 3×3 kernel with a stride of 1 and LeakyReLU activations is used, with residual scaling set to 0.2. The model operates in pretrained inference-only mode, using a ×4 PixelShuffle-based upsampling strategy.

Tesseract's LSTM-based pipeline is used by the OCR recognition module to decode sequences. A palm-leaf manuscript dataset with 1,086 samples from ten classic Tamil titles was used to train the integrated pipeline. Experiments were carried out by altering the number of epochs and batch sizes, while evaluating execution time and recognition accuracy, in order to assess the impact of training hyperparameters.

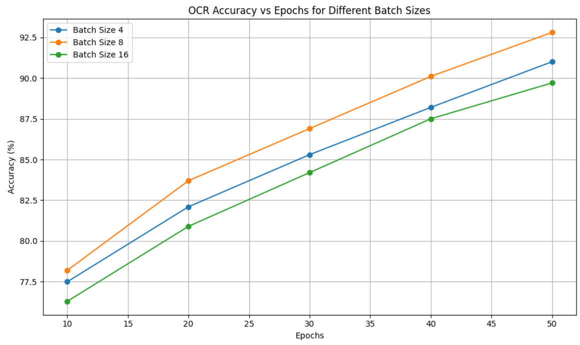

Figure 2: OCR performance under varying epochs and batch Sizes

The association between OCR accuracy and training epochs for different batch sizes is shown in Figure 2. The findings show that, in all configurations, character recognition accuracy steadily increases as the number of epochs increases. Interestingly, a batch size of 8 outperforms both smaller (batch size = 4) and larger (batch size = 16) options, achieving the highest identification accuracy of up to 92.8%. Batch size 4 has a higher computational cost even though it provides slightly better accuracy in the early epochs.

Epochs	Batch Size	Accuracy (%)	Execution Time (s)
10	4	89.2	34.7
20	4	91.3	46.1
20	8	93.5	40.3
30	8	**95.4**	58.2
40	16	94.1	71.8

Table 3: OCR performance under varying accuracy and execution time

On the other hand, Figure 3 illustrates that as batch size increases, execution time lowers. Larger batch sizes, like 16, in instance, result in quicker training periods, suggesting improved computational efficiency. However, the accuracy of recognition is slightly reduced as a result. These findings demonstrate that accuracy and execution time must be traded off, with a batch size of eight offering the best compromise for reliable and effective OCR on palm-leaf manuscripts.

C. Limitations and Future Scope

The algorithm may still encounter difficulties with manuscripts that have significant physical damage, significant text overlap, or missing character regions despite the noted improvements. To further improve recognition accuracy, future research can investigate including layout analysis, character-level segmentation, or transformer-based OCR models. Future developments will also take into account comparison analysis with other super-resolution models and quantitative evaluation utilizing character-level accuracy criteria.

VI. Future Enhancements

Enhancing the suggested Real-ESRGAN-assisted OCR framework's ability to adapt to multilingual scripts and diverse historical texts would be the main goal of future developments. By adjusting or enhancing super-resolution models to take into account script-specific stroke patterns in languages like Sanskrit, Telugu, Kannada, Malayalam, and Devanagari, as well as non-Indic characters found in archival records, the enhancement stage can be prolonged. The model's resilience to ink fading, bleed-through, and material deterioration will be further enhanced by integrating degradation-aware learning with artificially aged document datasets.

Future research will investigate the integration of transformer-based sequence models that can learn script-independent representations with language-adaptive OCR engines to provide multilingual recognition. This will allow for precise identification across many writing systems without requiring a great deal of manual modification. In order to manage complex historical formats, such as mixed scripts, annotations, and uneven text alignment, advanced document layout analysis and segmentation algorithms will also be introduced. The goal of these improvements is to develop the suggested framework into a generalized OCR solution for historical and multilingual document preservation, enabling extensive digitalization and cultural heritage preservation.

References:

[1] B. J. Bipin Nair, D. Arjun, S. Anandha Krishnan, T. Nandhakishor, A. R. Mohan, and V. Ravi, “Palm Leaf Manuscript Classification based on Degradation Level using Image Analysis Technique,” Proceedings IEEE, 2026. [Online]. Available: Acharya Institute of Technology via IEEE Xplore

[2] S. Sarithadevi and R. Rajesh, “Development of optimized ensemble machine learning-based character segmentation framework for ancient Tamil palm leaf manuscripts,” Engineering Applications of Artificial Intelligence, vol. 146, Art. no. 110235, Feb. 2025, doi: 10.1016/j.engappai.2025.110235.

[3] P. Bannigidad and S. P. Sajjan, “Historical Kannada Handwritten Palm Leaf (HKHPL): A Benchmark Dataset for Segmentation of HKHPL Manuscripts,” Cureus Journal of Computer Science, vol. 2025, no. 4, pp. 1–10, Aug. 2025, doi: 10.7759/s44389-025-04511-6.

[4] S. U. Maheswari, P. U. Maheswari, and G. R. S. Aakaash, “An intelligent character segmentation system coupled with deep learning based recognition for the digitization of ancient Tamil palm leaf manuscripts,” Heritage Science, vol. 12, art. no. 342, 2024, doi: 10.1186/s40494-024-01438-4.

[5] Mary Selvan and K. Ramar, “Development of optimized ensemble machine learning-based character segmentation framework for ancient Tamil palm leaf manuscripts,” Engineering Applications Artificial Intelligence, vol. 146, p. 110235, 2025, doi: 10.1016/j.engappai.2024.110235.

[6] E. Paulus, Jean Christophe Burie, and Fons J. Verbeek, “Supervised learning for low-resource isolated glyph recognition in palm leaf manuscripts,” Pattern Recognition, vol. 172, p. 112616, 2026, doi: 10.1016/j.patcog.2024.112616.

[7] Nimal Thuon, Jun Du, Panhapin Theang, and Ranysakol Thuon, “Multi-low resource languages in palm leaf manuscript recognition: Syllable-based augmentation and error analysis,” Pattern Recognition Letters., vol. 195, pp. 8–15, 2025, doi: 10.1016/j.patrec.2024.10.013.

[8] P. T. R. Pruthvi and B. J. Bipin Nair, "Hybrid image processing approach to enhance degraded palm leaf script," in Proceedings IEEE, 2024, pp. 1–5. Accessed: Jan. 13, 2026.

[9] Remya Sivan, Priyanka Prabhakar and Peeta Basa Pati, "Enhancing Generalization for Malayalam Palm Leaf Character Recognition through Cross-Data Stacking," International Conference on Power, Instrumentation, Control and Computing, 2025.

[10] Weihong Zhang, Xiaobo Li, Yizhao Huang, Shuping Xu, Junwu Tang, and Haofeng Hu, "Underwater image enhancement via frequency and spatial domains fusion," Optics and Lasers in Engineering, vol. 186, Art. no. 108826, 2025.

[11] Minglong Xue, Jinhong He, Wenhai Wang, and Mingliang Zhou, "Low-Light Image Enhancement via CLIP-Fourier Guided Wavelet Diffusion," ACM Transactions on Multimedia Computing Communications and Applications, vol. 21, no. 11, Nov. 2025, https://doi.org/10.1145/3764933.

[12] Zhen Zhu, Xiaobo Li, Qianwen Ma, Jingsheng Zhai, and Haofeng Hu, "FDNet: Fourier transform guided dual-channel underwater image enhancement diffusion network," Science China Technological Sciences, vol. 68, no. 1, pp. 1–17, Jan. 2025, doi: 10.1007/s11431-024-2824-x.

[13] Pallavi Singh, Ashish Kumar Bhandari, and Reman Kumar, "Low light image enhancement using reflection model and wavelet fusion," Multimedia Tools and Applications, vol. 84, pp. 24289–24317, 2025, doi: 10.1007/s11042-024-20086-1.

[14] Dan Xiang, Zebin Zhou, Wenlei Yang, Huihua Wang, Pan Gao, Mingming Xiao, Jinwen Zhang, and Xing Zhu, "A fusion framework with multi-scale convolution and triple-branch cascaded transformer for underwater image enhancement," Optics and Lasers Engineering, vol. 184, p. 108640, 2025, doi: 10.1016/j.optlaseng.2024.108640.

[15] Hao Wang, Kevin Köser, and Peng Ren, "Large foundation model empowered discriminative underwater image enhancement," IEEE Transactions on Geoscience and Remote Sensing, vol. 63, 2025, Art. no. 5609317.

[16] Mohd Ali, Mehboob Ali, Mubashir Hussain, and Deepika Koundal, "Generative adversarial networks (GANs) for medical image processing: Recent advancements," Archives of Computational Methods in Engineering, vol. 32, pp. 1185–1198, 2025, doi: 10.1007/s11831-024-10174-8.

[17] Zhen Zhu, Xiaobo Li, Qianwen Ma, Jingzheng Zhai, and Haofeng Hu, "FDNet: Fourier transform guided dual-channel underwater image enhancement diffusion network," Sci. China Technol. Sci., vol. 68, no. 1, pp. 1–17, Jan. 2025, doi: 10.1007/s11431-024-2824-x.

[18] Alexandru Brateanu, Raul Balmez, Adrian Avram, Ciprian Orhei, and Csomin Ancuti, "LYT-NET: Lightweight YUV Transformer-Based Network for Low-Light Image Enhancement," IEEE Signal Processing Letters, vol. 32, pp. 2065–2069, 2025.

[19] Noppol Anakpluek, Watcharakorn Pasanta, Latthawan Chantharasukha, Pattanawong Chokratsombat, Pajaya Kanjanakaew, and Thitirat Siriborvornratanakul, "Improved Tesseract optical character recognition performance on Thai document datasets," Big Data Research, vol. 39, 2025, https://doi.org/10.1016/j.bdr.2025.100508.

[20] Sankalp Nagaonkar, Augustya Sharma, Ashish Choithani, and Ashutosh Trivedi, "Benchmarking Vision-Language Models on Optical Character Recognition in Dynamic Video Environments," arXiv preprint arXiv:2502.06445, Feb. 2025. Available: https://github.com/video-db/ocr-benchmark.

[21] S. Nandhini Devi and N. Sabiyath Fatima, "Handwritten optical character recognition using TransRNN trained with self improved flower pollination algorithm (SI-FPA)," Multimedia Tools and Applications, vol. 84, pp. 19947–19969, 2025, Doi: 10.1007/s11042-024-19758-9.

[22] Çagri Sayallar and Ahmet Sayar, "Harmful ingredient detection from cosmetic products using

optical character recognition and bi-LSTM model," Signal, Image and Video Processing, vol. 19, p. 338, 2025, doi: 10.1007/s11760-025-03923-0.

[23] Carlos Penarrubia, Carlos Garrido-Munoz, Jose J. Valero-Mas, and Jorge Calvo-Zaragoza, "Spatial context-based Self-Supervised Learning for Handwritten Text Recognition," Pattern Recognition Letters, vol. 196, pp. 79–85, 2025.

[24] Qilong Li and Chongsheng Zhang, "UCR: A unified character-radical dual-supervision framework for accurate Chinese character recognition," Pattern Recognition, vol. 162, 111373, 2025.

[25] Ajit Singh, "Character Recognition in Ghibli-style Images: Investigating the Use of Deep Learning Techniques for Recognizing Characters," Social Science Research Network, Rochester, NY, SSRN Art. ID 5204852, Apr. 2025. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5204852.

[26] Carlos Penarrubia, Jose J. Valero-Mas, and Jorge Calvo-Zaragoza, "Self-Supervised Learning for Text Recognition: A Critical Survey," International Journal of Computer Vision, vol. 133, pp. 6221–6250, 2025, doi: 10.1007/s11263-025-02487-3.

[27] Yifei Zhang, Chang Liu, Jin Wei, Xiaomeng Yang, Yu Zhou, Can Ma, and Xiangyang Ji, "Linguistics-aware Masked Image Modeling for Self-supervised Scene Text Recognition," in Proceedings of IEEE/CVF Computer Vision Pattern Recognition (CVPR), 2025.