Comprehensive Evaluation of Tamil OCR Systems: A Survey, Dataset Creation, Benchmarking, and Error Analysis

S. Sivashanth, K. Sarveswaran, E. Y. A. Charles

Abstract— **This paper presents a comprehensive evaluation of Optical Character Recognition (OCR) systems for Tamil, a low-resource language that continues to trail behind high-resource counterparts in recognition accuracy despite substantial research efforts. Tamil OCR is inherently challenging due to the script's intricate character shapes, numerous ligatures, and its large set of 247 characters, including complex vowel-consonant combinations. These features complicate segmentation and recognition far more than Latin scripts. To contextualize the evaluation, a literature survey was conducted, covering key aspects such as pre-processing, recognition, and post-processing techniques. A major barrier in Tamil OCR research is the absence of a standardized diachronic benchmark dataset. To address this, we curated a dataset of 164 scanned images from printed documents spanning from 1850 to the present, taken at 10-year intervals. The collection captures eleven types of page layouts and eight document characteristics, considering noise levels, monolingual, and multilingual content, and printing technologies, etc. Expert-verified ground-truth data, including full-page transcriptions, line segmentations, and bounding boxes, enable detailed system evaluation. Using this dataset, we evaluated both commercial and open-source OCR systems through three strategies: full-page recognition, line-level segmentation, and bounding-box-based processing. Results show that while commercial systems perform better in terms of character and word accuracy, they struggle with complex layouts, degraded text, and historical typefaces. Although the focus is on Tamil, the evaluation approach and findings offer broader relevance for OCR research in other complex-script languages. The dataset and evaluation results are publicly available on GitHub to support future work in this domain.**

Keywords— **OCR, Benchmark Dataset, OCR Evaluation, Tamil, Pre-processing, Post-processing.**

I. Introduction

Natural Language Processing (NLP) has emerged as a pivotal area of Artificial Intelligence (AI) aimed at enabling computers and digital devices to understand, interpret, and generate human language. Essential resources for NLP include extensive text corpora for training and evaluation, benchmark datasets, lexicons, annotation tools, task-specific datasets, pre-trained models, and text-processing libraries.

Among these resources, Optical Character Recognition (OCR) plays a crucial role in digitizing printed and handwritten texts, forming the foundation for creating linguistic corpora essential for NLP tasks in low-resource languages, such as Tamil. The OCR converts text images into machine-readable and editable formats. OCR systems consist of various processing units to recognize characters, including the pre-processing phase, which primarily employs image-processing techniques to improve image quality and recognition accuracy, and the post-processing phase to refine the output [4].

OCR for Tamil presents inherent complexities due to the script's structural and historical characteristics. Tamil comprises 247 characters, including 12 vowels, 18 consonants, and 216 compound characters formed through vowel-consonant combinations, in addition to the unique character "ஃ" (Aytham). The script is highly agglutinative, with curved and visually similar glyphs, overlapping ligatures, and vowel markers that can appear before, above, or below the consonant base [5], [6]. These spatial and morphological features complicate character segmentation and recognition significantly more than in Latin-based scripts. Furthermore, the shapes of Tamil characters have evolved over time, even during the printing press era, due to regional typographic preferences, technological limitations, and orthographic reforms [7], [8]. This variability is compounded by the coexistence of multiple font encodings (e.g., Bamini, TAB, Unicode), making glyph standardization difficult [9], [10]. OCR development is further hindered by the lack of large-scale annotated datasets covering diverse fonts, scanning qualities, and historical styles [11]. Unlike scripts with case distinctions, Tamil also lacks capitalization cues, which reduces the availability of contextual features often used in post-processing. Altogether, these challenges necessitate tailored approaches in OCR model design and dataset construction to ensure reliable recognition across both modern and historical Tamil texts.

Tamil Optical Character Recognition is a specialized technology designed to recognize and convert Tamil text from scanned documents, images, or handwritten notes into machine-readable and editable formats. Research on Tamil Optical Character Recognition (OCR) has been notably fragmented, with various groups and individuals conducting studies independently at different times, leading to a lack of

Correspondence: Suthakar Sivashanth (E-mail: ssivashanth@univ.jfn.ac.lk)
Received: 11-09-2025 Revised: 08-01-2026 Accepted: 05-02-2026

Suthakar Sivashanth, Kengatharaiyer Sarveswaran, and Eugene Yugarajah Andrew Charles are from University of Jaffna, Sri Lanka (ssivashanth@univ.jfn.ac.lk,sarves@univ.jfn.ac.lk, charles.ey@univ.jfn.ac.lk)

DOI: 10.4038/icter.v18i3.7305

continuity and cohesion. This dispersion has resulted in duplicated efforts and slower progress in the development of robust standardized tools and datasets. Establishing a coordinated approach with collaborative efforts and shared resources is essential for enhancing the development and application of Tamil OCR technologies [12], [13]. To address these gaps, this survey paper provides a comprehensive overview of OCR techniques as well as pre-processing and post-processing methods specific to Tamil and other low-resource languages. This study references 20 research works related to Tamil OCR, identifying 13 OCR systems, of which only five are publicly available. This research also explored evaluation methods to effectively assess OCR performance. By highlighting the current state of Tamil OCR systems and identifying areas for future research, this survey aimed to bridge the gaps in Tamil OCR development and support broader NLP advancements in low-resource languages. This study exclusively addresses the challenges of digitizing printed Tamil texts, excluding the recognition of carvings such as stone and wood inscriptions and handwritten manuscripts such as ola and palm leaves. Although these forms are significant, they require distinct methodologies for effective recognition and preservation.

In addition to dispersed research efforts, the absence of a widely recognized benchmark dataset for the Tamil OCR further complicates the evaluation and comparison of different systems [14]. Although there are some available datasets, such as those curated by FutureBeeAI [15] and the Tamil OCR Object Detection Dataset by the IITB Research Work [16], they are not universally adopted, highlighting the need for a standardized dataset [17]. Different OCR systems rely on various types of datasets, including handwritten character and segmented word datasets. The lack of a common benchmark dataset hinders the standardized evaluation and comparison of system performance, further complicating progress in Tamil OCR. To fill this gap, this study focuses on developing a benchmark dataset. In addition, this paper presents the experimental evaluations of five OCR systems under diverse conditions, such as different page formats, image qualities, and processing methodologies.

The objectives of this study are as follows:

- Conduct a thorough review of OCR techniques reported in the literature, with a special focus on pre-processing, post-processing, and evaluation techniques that address the complexities associated with Tamil text recognition.
- Develop a benchmark Tamil OCR dataset that reflects the complexities of the printed Tamil script.
- Evaluate the performance of existing OCR systems for Tamil script in printed texts published after 1850.

The key contributions of this study are as follows:

- Comprehensive review: Conducted an in-depth analysis of OCR approaches, pre-processing, post-processing, and evaluation techniques.
- Dataset creation:
    - Compiled a benchmark image dataset comprising 164 pages from printed Tamil documents dating back to 1850, spanning diverse fonts, layouts, and printing conditions.
    - Annotated the dataset with verified ground truth text, ensuring high accuracy and consistency for OCR training and evaluation.
    - Performed line-level segmentation on selected pages, enabling fine-grained performance analysis for OCR models across different typesetting styles.
    - Generated bounding boxes for textual content, supporting spatial alignment and facilitating integration with modern OCR pipelines and machine learning workflows.
- Comparative evaluation:
    - Conducted a systematic evaluation of both commercial and open-source OCR systems using the developed benchmark dataset comprising printed Tamil texts from 1850 onwards.
    - Performed detailed error analysis to identify common failure modes, including character misrecognition, hallucination of non-existent symbols, segmentation failures (line, word, and layout-level), incorrect rendering of vowel modifiers and ligatures, mishandling of Tamil numerals, and inclusion of peripheral textual elements such as stamps and stickers.
    - Highlighted recurring challenges in handling the spatial and structural complexities of Tamil print, providing insights to inform future OCR research and system development.

The remainder of the paper is organized as follows. Section I introduces the significance of Optical Character Recognition for the Tamil language, the specific challenges associated with Tamil texts, the scope of the survey, and the objectives and contributions of the study. Section II presents a comprehensive literature review, covering character recognition approaches, state-of-the-art OCR systems, existing Tamil OCR systems, Tamil OCR datasets, pre-processing techniques, post-processing techniques, and evaluation methods relevant to Tamil OCR. Section III details the development of datasets, experiments conducted, results obtained from evaluating various OCR systems for Tamil, and error analysis. Finally, Section IV concludes the paper by highlighting the need for continued research in Tamil OCR and the broader impact of improved recognition systems on Tamil language users.

II. Literature Review

A. *Importance of OCR Systems for the Tamil Language*

Optical Character Recognition (OCR) is essential for preserving and revitalizing the cultural and historical heritage of a language. Many languages, including Tamil, have a rich literary tradition, with texts written on fragile materials such as paper, palm leaves, or stone inscriptions, which are vulnerable to decay over time. Digitizing these documents ensures their preservation in a durable format, safeguarding them against physical deterioration or loss, while enabling efficient information retrieval. OCR systems play a pivotal role in this process by converting printed texts into digital formats, making them accessible for future generations, and facilitating modern applications such as text classification, text summarization, language translation, information retrieval, sentiment analysis, and text-to-speech systems. Through these efforts, OCR not only preserves cultural heritage but also ensures broader accessibility and usability in the digital age.

In addition, most low-resource languages are from regions of developing or underdeveloped countries, where there is a lack of infrastructure and mechanisms to preserve books and other artifacts. Internal conflicts pose a threat to these resources, emphasizing the need for digitization. By digitizing texts, OCR systems help protect valuable cultural assets from physical damage and loss while also providing a means to preserve and share knowledge in regions where traditional preservation methods may be inadequate. This underscores the importance of OCR technology for safeguarding linguistic heritage and promoting global accessibility and understanding.

Tamil, a classical language [18], has a rich literary heritage spanning over two millennia. During this extensive period, Tamil evolved through various media and writing styles, reflecting the region's cultural, historical, and social changes. Numerous Tamil texts, including ancient stone inscriptions, palm-leaf manuscripts, and printed books still exist. This study focuses exclusively on the challenges associated with digitizing the printed Tamil texts. The recognition of carvings, such as stone and wood inscriptions, as well as handwritten manuscripts, such as ola and palm leaves, is considered outside the scope of this study. Although these forms are equally significant, they require distinct methodologies for their recognition and preservation.

Access to comprehensive historical data, including historical evolution and changes over time, is crucial for conducting a diachronic study of Tamil. Digitizing all Tamil texts, regardless of their form, is essential for preserving their cultural heritage and facilitating linguistic and literary research. The scarcity of digitized Tamil corpora poses a significant challenge for training Large Language Models (LLMs) [19], [20]. Consequently, LLMs often fail to handle Tamil language processing effectively, which includes understanding Tamil's heritage, knowledge systems, and cultural context [21]. Increasing the availability of digitized Tamil texts is critical for bridging this gap and empowering LLMs to support Tamil.

B. *Challenges of Tamil OCR*

Tamil texts have been captured in various forms throughout history, including carvings, handwritten manuscripts, and printed materials. Each medium presents unique challenges stemming from differences in the quality and techniques required for their preservation and digitization. For example, carvings on stones and wood often suffer from erosion and uneven surfaces, making their recognition difficult. Handwritten manuscripts, especially those on fragile palm or ola leaves, require careful handling and specialized imaging techniques to preserve clarity and detail. These challenges, particularly in handwritten Tamil, are further exacerbated by character shape variations and cursive styles, as highlighted in handwritten OCR studies [22], [23].

Adding to these challenges is the variation in Tamil writing styles over the centuries. Tamil character shapes have evolved significantly, and the structure of early writing is often unrecognizable to modern Tamil native speakers. This evolution not only complicates the recognition of historical texts but also requires contextual understanding for accurate interpretation. The Tamil script underwent a significant transformation over time. Fig. 1 provides a visual representation of the evolution of Tamil script over time. It includes various stages of development, showing how the script has transformed from its ancient forms to the modern script used today. Until the 20th century, Tamil literature was mostly written in poetic form, characterized by compact expressions and complex syntactical arrangements. These stylistic choices further add to the intricacies of recognizing and digitizing Tamil texts.

In addition to the structural and stylistic complexities, the lack of standardized font types and non-OCR-friendly font designs significantly hinders the accurate recognition of Tamil characters, posing a major challenge in Tamil OCR [24]. Furthermore, encoding Tamil symbols and numerals in Unicode presents another challenge. While basic Tamil letters are encoded in the Unicode Basic Plane [25] and additional numerals and symbols have recently been included in the Extended Plane [26], many newly encoded characters lack proper application and font support. Moreover, several symbols remain unencoded, limiting the ability to digitize very old Tamil documents comprehensively. The lack of standardized font encoding further exacerbates this problem, complicating the recognition and digitization processes [27].

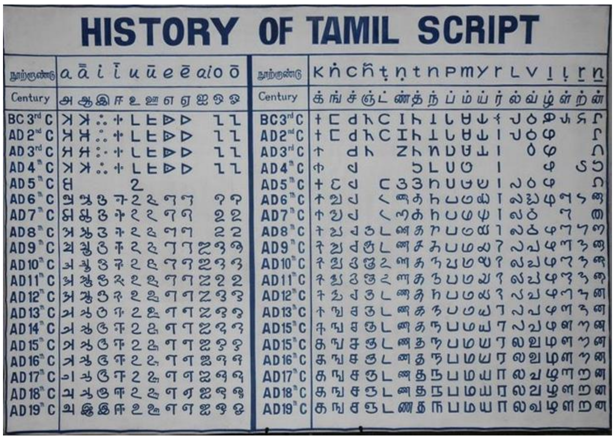


Fig. 1 Evolution of Tamil Script. Source: https://commons.wikimedia.org/wiki/File:History of Tamil Script.jpg

Building robust OCR systems for Tamil requires overcoming several obstacles. Previous research [27], [28], [29] has highlighted issues such as incorrect character recognition, difficulties in handling complex ligatures, low-quality or noisy images, and the processing of documents with intricate layouts. The complexity of the Tamil script, its large character set, intricate ligatures, and syllabic structure that incorporates compound characters and modifiers make the development of effective OCR models particularly challenging. Significant font variations, particularly in older or nonstandard typefaces, further complicate this task by preventing consistent text recognition across diverse printed materials.

The challenges of the Tamil script's complexity are compounded by the unique difficulties posed by the Tamil-Brahmi script, a precursor to modern Tamil. Its paleographic variations and orthographic intricacies require specialized methodologies and advanced machine-learning techniques to achieve high recognition accuracy [30]. Addressing these challenges requires a combination of tailored algorithms, comprehensive datasets, and contextual understanding of Tamil's linguistic and cultural evolution.

C. *Literature Summary on Character Recognition Approaches*

Character recognition has been the focal point of research on document analysis and OCR systems for several decades. The evolution of these approaches, especially for complex scripts such as Tamil, reflects broader technological advancements in the fields of pattern recognition and artificial intelligence. Early methods predominantly relied on handcrafted features and template matching, which, while foundational, were limited in their ability to handle script variation and noise.

The introduction of machine learning has led to significant advancements through techniques such as Support Vector Machines (SVM) and Hidden Markov Models (HMM), providing more flexible and robust models for recognizing diverse fonts and distorted characters. Recent advances in deep learning have transformed the field even further, with models like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) significantly enhancing the capabilities and accuracy of OCR systems.

In addition to these core approaches, modern advances, such as transfer learning, attention mechanisms, and Generative Adversarial Networks (GANs), have opened new possibilities, enabling higher accuracy and greater adaptability, particularly in resource-limited languages such as Tamil. These advancements address longstanding challenges by allowing for more efficient training processes and improved recognition of complex scripts.

1) *Traditional OCR Approaches:*

- Template Matching: Template matching is a traditional approach in which input characters are compared with predefined templates stored in the system [31].
- Statistical Methods: These methods use statistical properties of the characters, such as pixel distribution and geometric features, to recognize text. Techniques such as zoning and moment invariants fall into this category [32].
- Structural Methods: These methods focus on the structural features of characters, such as strokes, loops, and intersections. Graph-based representations and tree structures are often used to model the relationships between different parts of a character [33].
- Hybrid Methods: These combine elements of both statistical and structural methods to improve recognition accuracy. For example, a hybrid approach may use statistical methods for initial character segmentation and structural methods for final recognition.

These traditional methods laid the groundwork for more advanced machine-learning and deep-learning techniques. Although these methods are straightforward and work well for printed characters with consistent fonts, they face significant challenges in recognizing handwritten text and variations in font styles commonly found in Tamil scripts. A similar limitation is observed in recognizing artistic Chinese calligraphy, where traditional methods fail to handle the complex structures and diverse styles effectively [34].

2) *Machine Learning-Based Approaches:* Feature-Based Character Recognition

In machine learning-based optical character recognition (OCR), feature-based approaches play a crucial role in identifying and classifying characters. These methods involve extracting distinctive attributes from character images, such as shape, texture, or pixel distribution, which serve as inputs to learning algorithms. By focusing on these features, models can generalize across variations in font, size, and noise, improving recognition accuracy. Common features used in these approaches include:

- Geometric Properties: Aspect ratio, curvature, and stroke width help differentiate character shapes.
- Texture-Based Features: Gradient histograms and local binary patterns capture intricate details and patterns within the character structure [35].
- Statistical Features: Pixel intensity distributions and moments provide insights into the overall appearance and density of characters [36].
- Frequency-Domain Features: Derived from Fourier or wavelet transforms, these features analyze spatial frequency components, enhancing recognition capabilities [37].
- Histogram of Oriented Gradients (HOG): HOG features capture edge directions and are particularly effective for character recognition tasks due to their robustness to variations in illumination and pose.
- Scale-Invariant Feature Transform (SIFT): SIFT is used to detect and describe local features in images, robust to changes in scale, rotation, and illumination.
- Speeded-Up Robust Features (SURF): SURF is a faster alternative to SIFT, designed for real-time applications, and robust to various transformations.
- Local Binary Patterns (LBP): LBP is a texture descriptor that labels an image's pixels by thresholding each pixel's neighborhood and considering the result as a binary number. It captures local texture and structure of characters, making it effective in distinguishing different shapes and styles, especially in varying font and handwriting scenarios.
- Gabor Filters: Gabor filters are used for texture analysis and feature extraction. They analyze the specific frequency content in an image at various directions and scales, similar to how the human

visual system perceives textures. In OCR, they enhance recognition of complex and distorted text [38].

- Edge Orientation Histograms: These histograms capture the distribution of edge orientations in an image. This feature is robust to variations in illumination and pose, aiding accurate identification and classification of text under diverse conditions.

Once relevant features are extracted from character images, various machine learning algorithms are employed to classify and recognize the characters. These algorithms learn patterns from the feature representations and make predictions based on learned associations. The choice of algorithm often depends on the nature of the features, the size of the dataset, and the desired trade-off between accuracy and computational efficiency. Algorithms commonly used in machine learning-based approaches include:

- Support Vector Machines (SVMs): SVMs determine an optimal hyperplane to separate different character classes. They have been successfully applied to Tamil OCR, achieving high accuracy rates [22], [39], [40], [41], [42]. SVMs are particularly effective in high-dimensional spaces and are versatile with different kernel functions such as linear, polynomial, and radial basis function (RBF) kernels.

- Hidden Markov Models (HMMs): HMMs are particularly useful for sequential character recognition tasks, making them suitable for recognizing handwriting or cursive scripts. They model the probability of sequences of observed events, which is ideal for capturing the temporal dependencies in handwriting [43].

- Statistical Methods: Techniques such as k-nearest neighbours (k-NN) and Bayesian classifiers analyze the statistical properties of the extracted features to classify characters. k-NN is a non-parametric method that classifies based on the majority vote of the nearest neighbors, whereas Bayesian classifiers apply Bayes' theorem to predict the probability of a character belonging to a particular class [44].

- Random Forests: Random forests are an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes for classification tasks. They are robust to overfitting and can handle large datasets with high dimensionality.

Table I provides a summary of highly cited papers on feature-based OCR approaches, detailing the features used, datasets, and algorithms applied in these studies.

While feature-based OCR methods have shown good performance for certain languages and specific tasks, they have limitations, such as the need for manual feature engineering and difficulty in handling complex and noisy data. Deep-learning approaches address these challenges by automatically learning and extracting relevant features from raw pixel data, making them highly effective for diverse OCR applications [47]. They handle complex patterns, noisy data, and varying document qualities, thereby providing superior accuracy and robustness [48]. For example, ensemble deep convolutional neural networks have shown excellent performance in recognizing handwritten Tai Le characters, a challenging script with high character similarity and variability [49]. In addition, deep learning models offer scalability and flexibility, allowing them to be fine-tuned for specific tasks or languages [50]. Deep-learning-based OCR can also be integrated with other AI technologies, such as Natural Language Processing (NLP) and computer vision, to provide a more comprehensive document understanding and

TABLE I
HIGHLY CITED PAPERS ON FEATURE-BASED OCR APPROACHES

Research Paper	Features Used	Data Set Used	Algorithm Used
Gradient-Based Learning Applied to Document Recognition [37]	Gradient-based features	MNIST	Convolutional Neural Networks (CNNs)
Gabor Filters for Document Image Analysis [35]	Gabor filters	Custom dataset of handwritten characters	Support Vector Machines (SVMs)
Dissimilarity Representation for Pattern Recognition [36]	Statistical features (pixel intensity distributions, moments)	Custom dataset of printed characters	k-Nearest Neighbors (k-NN)
Indian Script Character Recognition: A Survey [39]	Geometric properties	Indian language dataset	Hidden Markov Models (HMMs)
Tamil OCR Using Support Vector Machines [40]	Texture-based features (gradient histograms, local binary patterns)	Tamil OCR dataset	Support Vector Machines (SVMs)
Bayesian Classifiers for Printed Character Recognition [41]	Frequency-domain features (Fourier transforms)	Custom dataset of printed characters	Bayesian classifiers
Printed Tamil Character Recognition Using HOG Features [42]	Histogram of Oriented Gradients (HOG)	Custom dataset of printed Tamil characters	Support Vector Machines (SVMs)
Scene Text Recognition using Co-occurrence of Histogram of Oriented Gradients [45]	Co-occurrence histogram of oriented gradients (Co-HOG)	ICDAR2003 and Street View Text (SVT) datasets	Support Vector Machines (SVMs)
Handwritten English Character Recognition Using SIFT [46]	SIFT features	English Handwritten Dataset	Random Forests

analysis [51]. Furthermore, deep learning approaches have shown promise in improving the OCR performance for low-resource languages by leveraging transfer learning and multilingual models [52]. These advantages make deep learning a promising direction for advancing OCR technology.

3) *Deep Learning Approaches:*

- Convolutional Neural Networks (CNNs): CNNs have revolutionized OCR by enabling systems to learn features directly from raw data without requiring manual feature extraction. They automatically learn features such as edges, textures, and shapes, making them highly effective for recognizing printed and handwritten texts [53]. Recent studies [14] demonstrated that CNNs perform exceptionally well in Tamil OCR tasks, effectively handling script complexities, including varying font styles and intricate ligatures, with high accuracy.

- Recurrent Neural Networks (RNNs): RNNs, particularly Long Short-Term Memory (LSTM) networks [54], [55], excel at recognizing sequences of characters, making them particularly useful for OCR tasks involving continuous scripts. They help recognize text sequences and handle text lines of varying lengths. By capturing the contextual relationships between characters, RNNs have significantly improved the accuracy of OCR for complex texts.

4) *Recent Advances in OCR:*

- Transfer Learning: Transfer learning [56] has been increasingly used to improve OCR accuracy, especially for languages with limited training data such as Tamil. By fine-tuning pre-trained models on large datasets, researchers have been able to significantly boost recognition rates for challenging scripts while reducing the need for large amounts of domain-specific data.

- Attention Mechanisms: Attention mechanisms have brought a new level of precision to OCR systems by allowing models to focus on specific parts of an input image [57]. This helps in accurately recognizing complex or noisy characters in scripts, such as Tamil, where characters may vary in size and form. Recent advances also include combining attention with super-resolution techniques, such as the CNN-based batch-transformer network (BT-STISR), which uses self-attention and global attention to enhance low-resolution text images for better recognition accuracy [58].

- Generative Adversarial Networks (GANs): GANs are gaining attention for their ability to generate synthetic training data that can be used to augment OCR datasets. This approach is particularly valuable for Tamil OCR, where creating a comprehensive dataset may be challenging. By generating additional training samples, GANs can improve the models robustness and accuracy in real-world scenarios [59].

D. *State-of-the-Art OCR systems*

Recent studies have highlighted the effectiveness of OCR systems in recognizing printed and handwritten texts across different scripts. Among the currently best OCR systems, Tesseract OCR, Amazon Textract, and Google Document AI stand out for their high accuracy and versatility. Tesseract OCR, developed by Google, is widely used in academic research and practical applications owing to its robust performance and support for multiple languages [60]. Amazon Textract excels in extracting text from complex documents, including forms and tables, using machine-learning models, and Google Document AI leverages advanced AI to provide high accuracy, particularly in noisy document conditions, making it a strong contender in the OCR field [61].

OCR technology has advanced significantly, achieving high accuracy rates in various languages, including low-resource languages. For example, an OCR system developed for Marathi achieved a character-level accuracy of 88% in recognizing printed documents [62]. Similarly, a Chinese-English mixed OCR system demonstrated a character-level error rate of 0.87% for magazine samples and 0.75% for book samples [63].

E. *Existing Tamil OCR systems*

Several Optical Character Recognition (OCR) systems have been developed for Tamil, showcasing the diverse methodologies and approaches that have significantly contributed to this field. However, challenges remain in accurately recognizing specific characteristics and ligatures, underscoring the need for further improvement.

Previous studies have reported that, although these OCR systems demonstrate promising results, they continue to face significant challenges. These include misidentification of Tamil numerals, confusion among similarly shaped characters, difficulties with code-mixed text, low performance on poor-quality images, and documents with complex layouts [27], [28], [64].

To provide more detailed insights, Table II summarizes the key features, datasets, and evaluation metrics of the various Tamil OCR systems. This comparison highlights the strengths and limitations of each system, offering a comprehensive overview of the progress made in the Tamil OCR and the gaps that remain to be addressed. The accessibility of these OCR systems is stated along with their names in a table.

F. *Tamil OCR Datasets*

Tamil documents come in various forms, each of which presents a unique challenge for OCR systems. These include printed texts, such as books, newspapers, magazines, and pamphlets, which vary in font style, size, and layout. Handwritten texts encompass personal notes, letters, and forms and present a wide range of handwriting styles and quality. Historical manuscripts, including ancient scripts and palm-leaf documents, often suffer from wear and tear, further complicating OCR tasks.

Although there are a few specialized datasets available, such as the uTHCD dataset [14], which provides approximately 91,000 samples for handwritten Tamil character recognition, comprehensive datasets for Tamil OCR, particularly for printed text and historical documents, are still scarce. Recently, a new dataset specifically designed for word-level recognition of Indic languages, including Tamil, was introduced [71]. This dataset includes training, testing, and validation splits, with images annotated at the word level. The word-based segmentation approach makes it particularly well-suited for developing and evaluating OCR models aimed at Tamil word recognition. Such resources significantly enhance the capability to train robust OCR systems and address challenges related to font variations, noise, and complex layouts.

This growing collection of datasets, including both character-level and word-level resources, paves the way for more effective and adaptable OCR models for the Tamil language processing.

TABLE II
COMPARISON OF TAMIL OCR SYSTEMS

References	Datasets/Technologies/Evaluation
Aharamariyi Tamil OCR [27] (Not accessible)	**Used Datasets:** 501 words from a small Tamil corpus **Used Technologies:** Tesseract engine **Evaluation:** Character-Level Accuracy 81%
Tamizhi-Net OCR [28] (Not accessible)	**Used Datasets:** PDF documents from the official websites of the Sri Lankan parliament **Used Technologies:** Tesseract engine, LSTM-based training **Evaluation:** Reduced character-level error rate of Tesseract to 2.61% for Tamil and to 4.74% for Sinhala. The word-level error rates were reduced to 20.61% for Tamil and 26.58% for Sinhala.
OCR Tamil [65] (Accessible)	**Used Datasets:** Did not explicitly mention the dataset **Used Technologies:** Permuted autoregressive sequence (PARSeq) models, CRAFT text detector **Evaluation:** Character-Level Accuracy 95% for newly printed Tamil books
Hybrid Decision Tree-based OCR System [66] (Not accessible)	**Used Datasets:** Tested on a dataset of 12,400 Tamil character samples **Used Technologies:** Hybrid approach of DAG and UDT SVMs with a radial basis function (RBF) kernel **Evaluation:** Character-Level Accuracy 98.80%
A Multi-Font, Multi-Size Optical Character Recognizer (OCR) of Tamil [67] (Not accessible)	**Used Datasets:** Symbols extracted from Tamil texts in printed books; rare symbols supplemented with computer-generated fonts **Used Technologies:** Skew correction using Hough Transform and PCA; morphological processing; connected component analysis; three-level tree-structured classifier with nearest neighbor using Euclidean distance **Evaluation:** Character-Level Accuracy 99.1%
Tamil GNANI [68] (Not accessible)	**Used Datasets:** Over 4,000 samples, including Tamil text from magazines, novels, technical papers, and shloka books. Fonts such as TM-TT Valluvar, TAB_Arulmathi, Inaimathi, TM-TT Bharathi, and TAM-Aniezhai. Tested on font sizes 14 to 20 **Used Technologies:** Pre-processing (binarization, skew detection), segmentation (projection profiles, connected component analysis), normalization, feature extraction (Block DCT), Nearest Neighbor Classifier, implemented in Visual C++ **Evaluation:** Character-Level Accuracy 98%
Omnifont Tamil OCR [42] (Not accessible)	**Used Datasets:** Annotated corpus of 1,000 scanned pages from books printed between 1950–2002, XML-annotated database with 5,000 scanned pages and Unicode-typed text **Used Technologies:** Karhunen-Loeve Transform (KLT) features, Support Vector Machine (SVM) classifier with RBF kernel, Discriminative Directed Acyclic Graph (DDAG) configuration, noise-tolerant segmentation **Evaluation:** Character-Level Accuracy 94% for Tamil
OCR software for printed Tamil text [69] (Not accessible)	**Used Datasets:** Neatly printed Tamil text, old books **Used Technologies:** Interactive recognition, training module, spell-check integration **Evaluation:** Character-Level Accuracy over 99% for neatly printed Tamil text
A Complete OCR System for Tamil Magazine Documents [70] (Not accessible)	**Used Datasets:** Scanned Tamil magazine pages **Used Technologies:** Pre-processing (compression, skew correction using CSP, Otsu's binarization, noise removal), block segmentation and classification using Radial Basis Function Neural Networks (RBFNN), character recognition using RBFNN with Gabor filters, connected component extraction, reconstruction into HTML format **Evaluation:** Character-Level Accuracy 90-97%

G. Pre-processing Techniques for Tamil OCR

Pre-processing plays a crucial role in enhancing the accuracy of the OCR. Techniques such as image binarization, noise reduction, and skew correction are essential for optimizing inputs for OCR systems. [72] introduced a skew detection method utilizing the Radon transform, achieving a recognition rate of 96.30% for skew correction. They also proposed an SVM-based classification technique for distinguishing between text and non-text regions, reporting a classification accuracy of 99.18%. Addressing issues such as lighting, rotation, and resolution in scanned image quality are crucial, as these factors significantly affect recognition performance. A multithreaded approach to pre-processing tasks, as proposed in [73], has shown potential in reducing computational time by processing skew detection, binarization, and segmentation tasks in parallel threads, thereby enhancing efficiency. Font variations pose a significant challenge for OCR systems, particularly for complex scripts, such as Tamil. Embedding optical co-features into fonts, as proposed in [24], offers a novel solution to enhance machine readability without compromising human legibility.

H. Post-processing Techniques for Tamil OCR

Post-processing plays a crucial role in improving the output accuracy of the Tamil OCR systems. Various techniques have been explored to refine raw OCR output, enhancing both syntactic and semantic correctness. One of the most prominent approaches involves using a tree-based algorithm to identify missing or incorrect words, thereby ensuring that the detected words maintain their contextual relevance. In addition, stemming and lemmatization have been employed to reduce words to their root forms, which guarantees semantic accuracy, a method that has proven effective in Tamil OCR systems [74].

Although these techniques address syntactic accuracy, a deeper contextual refinement requires more sophisticated approaches. Beyond semantic refinement, rapid error correction tools such as SymSpell have been utilized to speed up the detection and rectification of errors in OCR outputs. SymSpell offers quick correction suggestions, whereas Bloom filters serve as validation tools to ensure that only valid words are retained, thereby adding an extra layer of error prevention [75]. These techniques help streamline the OCR correction process and ensure a cleaner output.

To achieve deeper contextual accuracy in Tamil OCR, word embeddings [76] and large language models (LLMs) have emerged as promising tools for addressing grammatical and semantic inconsistencies by capturing the language's complex linguistic nuances. Notably, a recent approach [77] demonstrated this potential by employing a multilingual RoBERTa model to develop a context-sensitive Tamil spellchecker, achieving over 91% accuracy in error detection - highlighting the broader applicability of LLMs to post-OCR correction tasks, particularly for long-form documents where context is critical [19].

By integrating LLMs, OCR systems can improve their understanding of text structure and meaning, resulting in more accurate corrections, particularly for tasks such as Named Entity Recognition (NER), which identifies proper nouns and domain-specific terms. However, challenges remain, particularly the limited availability of Tamil-specific LLMs and the substantial computational resources required for large-scale applications. Despite these challenges, LLMs offer a promising future for enhancing OCR systems by addressing complex linguistic nuances and improving the document-level coherence.

Canti Error Detection is another crucial method that ensures the correct use of vowel modifiers, which are often misrecognized in Tamil scripts. Maintaining correct vowel modifiers is essential for preserving the phonetic and visual integrity of a text, contributing to both its legibility and accuracy. To complement these error detection mechanisms, word frequency analysis and character-level bigram similarity have been applied to optimize corrections by prioritizing frequent words and correcting improbable character combinations [78].

Statistical models, such as n-gram models, have also been applied to correct OCR errors by predicting the likelihood of certain characters and word sequences in Tamil text. For instance, the use of bigram models improves the accuracy of OCR by leveraging statistical predictions to correct common errors in word sequences [79]. These models help minimize improbable word combinations, thereby increasing the reliability of OCR outputs [80].

Spell checkers generally address two categories of errors: non-word and true-word errors. Non-word errors may arise from invalid words or words that are valid but absent from a given lexicon. True word errors, on the other hand, refer to words that are valid in isolation but contextually inappropriate within a sentence. Reference [81] demonstrates that combining bigram probabilistic models with Minimum Edit Distance (MED) techniques can effectively detect and correct such real-world errors in Tamil. This hybrid approach has shown promising results in improving contextual accuracy, making it particularly beneficial for OCR post-processing and broader natural language processing tasks.

In addition to statistical approaches, rule-based correction techniques have been applied to the Tamil OCR systems, incorporating Tamil-specific logistic rules to correct common errors. This includes handling vowel modifiers, compound characters, and sandhi rules that govern the sound combinations between words in Tamil. Such rules are essential for aligning OCR outputs with the unique linguistic features of Tamil [82].

Finally, machine-learning approaches have shown potential in post-processing Tamil OCR systems. Techniques such as Statistical Machine Translation (SMT) and Neural Machine Translation (NMT) have demonstrated their ability to learn correction patterns from large datasets, where parallel corpora of OCR outputs and their manual corrections are used to refine system performance. These machine learning methods enable more dynamic corrections that improve over time as the model is exposed to more data, thus enhancing the accuracy and adaptability of the OCR system.

I. Evaluation Techniques

Standardized benchmarks are essential for comparing the performance of different Tamil OCR systems. Common evaluation metrics include Character Error Rate (CER), which measures the edit distance between the OCR output and ground truth at the character level [83]. CER is the fraction of the sum of the number of substitutions, deletions, and insertions, considered to the total number of characters in the expected output. The Word Error Rate (WER) provides a more semantic evaluation by calculating the edit distance at the word level and assessing the accuracy of OCR output in capturing meaningful text units [84]. In WER, substitutions, deletions, or insertions are considered when creating the sentence, as in the expected output [85].

To further enhance the evaluation accuracy, methods such as the Levenshtein distance and its variant, that is, the Modified Levenshtein Distance [86], have been utilized. In these approaches, substitutions, deletions, and insertions are assigned specific weights based on character similarity, providing a more nuanced measurement of the edit distance between the OCR output and the ground truth. This adaptability is particularly useful when assessing the performance across different Tamil scripts and fonts. The unique word anchor method further contributes to the evaluation by identifying keywords in the text, enabling more robust comparisons of significant textual components.

Another approach, the Longest Common Substring Method, proved valuable by identifying the longest contiguous sequence of characters shared between the OCR output and the ground truth [87]. This method provides insight into a system's ability to preserve text continuity and mitigate the impact of isolated errors. Combined with standard evaluation metrics, these techniques offer a comprehensive assessment of OCR systems, contributing to their continuous development and refinement in Tamil-language document processing.

Additionally, the F1 Score, which balances precision and recall, offers a comprehensive measure of the OCR performance across various datasets and document types. The F1 Score can be calculated at both character and word levels. These metrics play a pivotal role in evaluating the reliability and effectiveness of OCR systems in Tamil documents.

III. Evaluation of the Existing OCR systems

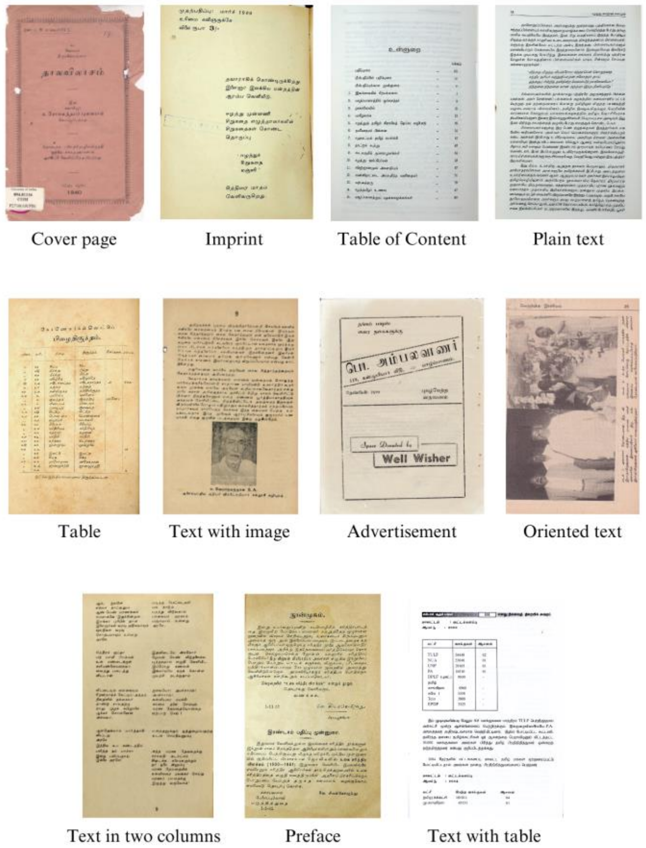
A. Dataset Development

The dataset for this study was developed using a collection of 164 images sourced from the Library, University of Jaffna, and Noolaham Foundation, covering publications from 1850 and selected at 10-year intervals. The Library, University of Jaffna, is a major academic information resource centre in Northern Sri Lanka that supports extensive collections across various disciplines [88]. The Noolaham Foundation is a nonprofit organization dedicated to digitizing and preserving documents related to Tamil-speaking communities, offering a vast and diverse archive [89]. Both institutions boast extensive collections of documents to ensure that the dataset encompasses a diverse set of images. These images represent various document types, including cover pages, imprints, tables of content, plain text, tables, text accompanied by tables or images, advertisements, oriented text, and text in two columns, as shown in Fig. 2. This diversity is crucial for capturing a wide range of contexts, thereby enhancing the robustness, comprehensiveness, and adaptability of Tamil OCR system evaluations.

Each image was categorized on the basis of three primary attributes: image condition, mono/multilingual text, and printing technology. A knowledgeable expert in historical printing technologies was consulted to verify the specific methods used in each case.

- **Image Condition:** Images were classified as good, damaged, or noisy, reflecting the preservation state and quality of the materials, which can significantly impact OCR recognition accuracy.
- **Number of Languages:** Each document was labeled as either monolingual or multilingual, depending on whether the images contained text in a single language or a combination of multiple languages.
- **Printing Technology:** The printing technique was identified as either a letterpress or digital print, distinguishing historical prints from modern reproductions and highlighting technical challenges for OCR systems with different types of printing.

The dataset encompasses a diverse range of sources, including books, newspapers, magazines, and pamphlets, ensuring a broad spectrum of textual forms and structures. To maintain high-quality inputs suitable for OCR tasks, all images were scanned at a resolution of 300 dots per inch (DPI), providing the necessary clarity to enable accurate character recognition.



Cover page Imprint Table of Content Plain text

Table Text with image Advertisement Oriented text

Text in two columns Preface Text with table

Fig. 2 Examples of diverse page formats.

Ground-truth text was manually created for each image with the support of the Google Docs OCR engine to ensure accuracy. Because Google Docs does not support TIFF image files, the images were first converted into the PNG format. The converted images were then processed using Google Docs to extract text, which was subsequently manually corrected with the assistance of linguistic experts. This ground truth was used as a reference to evaluate the OCR outputs, enabling quantitative comparisons between the recognized text and ground truth.

To facilitate structured evaluation, a detailed metadata file was created to capture the essential attributes of each image. The metadata file was organized into three primary fields.

- **Image Information:** Specifies image format, resolution, memory size, paper color, image condition, width, and height.
- **Data Information:** Lists attributes such as font size, font style, resource type, script, number of languages, domain, genre, and content format.
- **Data Source Information:** provides contextual information about the document's origin and historical background, essential for traceability, including its title, subtitle, alternative title, parallel title, author, publisher, edition, publication year, printing location, type of material, printing technology, ISBN/ISSN, and copyright details.

Using the metadata file, images can be filtered based on specific characteristics such as the publication era, document type, and/or script language. This allows for targeted evaluation of OCR systems under various conditions.

In addition, bounding boxes were manually created for 164 images, covering all diverse page formats, image conditions, printing technologies, and language types present in the dataset. Both line- and region-based approaches were used to create bounding boxes by leveraging software called LabelImg [90] to annotate the images. The line-based approach is particularly useful because of varying font styles and text sizes. The bounding boxes for each image were saved as JSON files, and each bounded text was manually annotated with the bounding-box coordinates (xmin, ymin, xmax, and ymax) to establish the ground truth. The ground-truth data were integrated to create a benchmark dataset for experiments involving bounding boxes. Additionally, five plain-text images were randomly selected from the dataset and cropped line-by-line, with each cropped section saved individually to evaluate the line-by-line recognition performance of the systems.

The resulting dataset provides a comprehensive foundation for testing the effectiveness and efficiency of OCR systems in Tamil texts, fostering advancements in technology designed to handle the complexities of Tamil scripts across diverse document types and historical contexts.

B. *Scanning Challenges*

Scanning historical documents poses several challenges. The key difficulties encountered include the following.

- **Lighting Issues:** Uneven lighting during scanning caused unwanted shadows on the pages, affecting text clarity and introducing noise that complicated OCR recognition.
- **Page Fragility:** Many documents, particularly those from the 19th and early 20th centuries, had delicate and brittle pages due to paper degradation over time. Special care is required to handle these materials without tearing or causing further damage during scanning.
- **Aging Effects:** Most of the scanned pages had turned yellowish over time due to the natural aging process of paper. This discoloration, combined with faded or degraded text, poses an additional challenge for OCR systems to accurately distinguish the characters from the background.
- **Distorted or Warped Pages:** Some books were tightly bound, causing text near the margins to appear curved or distorted when scanned. This issue makes it more difficult for OCR models to recognize text accurately without additional pre-processing.

C. *Experiments*

The experiments conducted in this study aimed to evaluate the performance of five OCR systems: Google Cloud Vision API (GCV API) [91], Google Docs API (GD API) [92], open-source Tesseract OCR (OST OCR) [60], Surya OCR [93] and OCR Tamil [65]. These systems were selected to represent a diverse set of approaches, including both commercial and open-source solutions. Google Cloud Vision and Google Docs APIs are widely recognized commercial tools known for their high accuracy and multilingual support. The Tesseract OCR was chosen as the baseline open-source engine given its widespread adoption, active development community, and comprehensive language support, including Tamil. Surya OCR and OCR Tamil were chosen due to their public availability.

To thoroughly assess the strengths and limitations of these systems, three distinct evaluations were performed to gain a comprehensive understanding of their capabilities, focusing on how well these systems handle challenges, such as noisy and damaged images, diverse text formats, multilingual documents, variations in printing technologies, and diverse page layouts.

The OCR quality assessment was conducted using the Character Error Rate (CER) and Word Error Rate (WER), which are the standard metrics in OCR evaluation. CER, defined in Equation (1), measures the number of character-level insertions, deletions, and substitutions required to match the OCR output to the ground truth, normalized by the number of characters in the ground truth. Similarly, WER, defined in Equation (2), evaluates errors at the word level, capturing the effectiveness of the OCR system in preserving the semantic integrity of the text.

$$\mathrm{CER} = (S + D + I) / N \tag{1}$$

Where:

- S = Number of character substitutions
- D = Number of character deletions
- I = Number of character insertions
- N = Total number of characters in the ground truth

$$\mathrm{WER} = (S + D + I) / N \tag{2}$$

Where:

- S = Number of word substitutions
- D = Number of word deletions
- I = Number of word insertions
- N = Total number of words in the ground truth

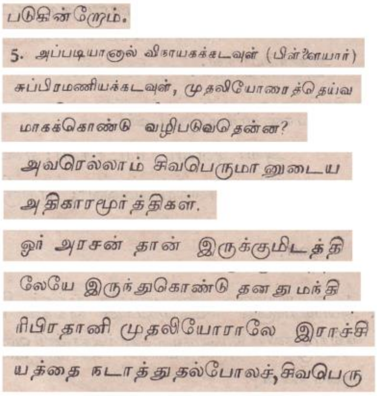
While CER is widely used, it assigns the same penalty to all substitutions, insertions, and deletions. However, in Tamil, some characters are visually or linguistically similar, such as ‘க்’ and ‘க’, and their confusion reflects a minor recognition error. Conversely, confusing ‘க்’ with a visually and phonologically different character like ‘ல்’ is more severe. CER does not distinguish between these cases, potentially underestimating the true severity of certain OCR errors. Future work could incorporate weighted error metrics that assign higher penalties to dissimilar characters and lower penalties to closely related vowel- or consonant-modified forms.

1) *Evaluation - Performance for diverse page formats and special categories:*

The first experiment assessed the ability of the OCR systems to recognize Tamil text across different page formats and conditions. These formats include cover pages, imprints, prefaces, tables of contents, plain text, text with images, text with tables, multi-column layouts, oriented text, and advertisements. The evaluation also considered different image conditions, good, damaged, noisy, and noisy-damaged, along with monolingual and multilingual texts. Additionally, documents printed using letterpress and digital printing technologies were included to assess system performance across different printing methods. This experiment was designed to explore insights into how each OCR system handles diverse page formats and conditions, highlighting its ability to accurately recognize Tamil text.

2) Evaluation - Performance for line-by-line segmented text images:

The second experiment compared the performance of OCR systems using two distinct processing methods: line-by-line segmentation and full-image processing. In the line-by-line approach, images are segmented into individual lines while maintaining the same resolution, with each line processed separately before merging the results into the final recognized text. An example of a line-by-line segmented image is shown in Fig. 3. This evaluation examined the benefits of dividing images into smaller sections, particularly when handling text alignment, noise, and segmentation issues, as opposed to processing an entire image in one pass.



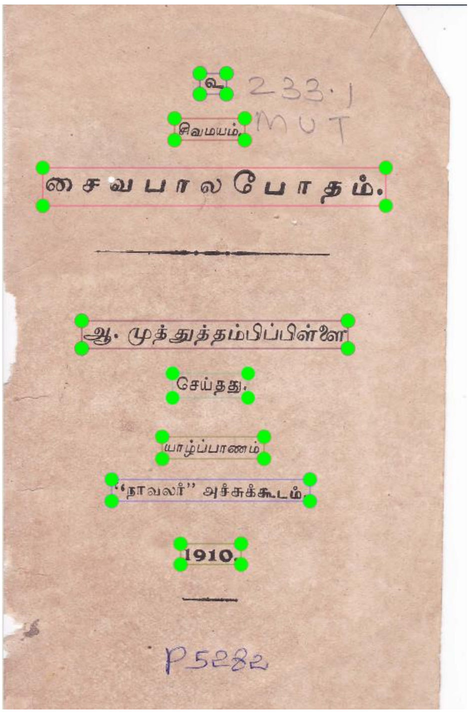
படுகின்றோம்.
5. அப்படியானால் விநாயகக்கடவுள் (பிள்ளையார்)
சுப்பிரமணியக்கடவுள், முதலியோரைத்தெய்வ
மாகக்கொண்டு வழிபடுவதென்ன?
அவரெல்லாம் சிவபெருமானுடைய
அதிகாரமூர்த்திகள்.
ஓர் அரசன் தான் இருக்குமிடத்தி
லேயே இருந்துகொண்டு தனது மந்தி
ரிபிரதானி முதலியோராலே இராச்சி
யத்தை நடாத்துதல்போலச், சிவபெரு

Fig. 3 Line-by-line segmentation. Each line is extracted and processed separately to enhance recognition accuracy.

3) Evaluation - Performance for bounding-box defined images:

The third experiment focused on bounding-box segmentation to assess OCR performance. In this approach, images are segmented into predefined regions based on bounding boxes, and text recognition is performed separately within each box. An example of bounding-box segmentation is shown in Fig. 4. The results from each cropped section were compared against the corresponding ground truth text, with bounding-box coordinates included for precise comparison. This experiment aimed to evaluate the effectiveness of the systems in handling text extracted from specific regions of an image and to understand how bounding-box detection influences OCR accuracy. It also provides insights into the challenges and advantages of isolating text sections for OCR, particularly in documents with complex layouts or overlapping content.

Each of these experiments was designed to obtain valuable insights into the performance of OCR systems under various conditions, thereby providing a deeper understanding of their strengths and limitations. These evaluations are expected to highlight the trade-offs between different processing methods and shed light on the factors that influence the accuracy of Tamil text recognition across diverse document types and image conditions. The outcomes of these experiments are anticipated to identify the most effective approaches for optimizing OCR performance in Tamil text recognition.



உ
233.1
சிவமயம்.
MUT
சைவபாலபோதம்.
ஆ. முத்துத்தம்பிப்பிள்ளை
செய்தது.
யாழ்ப்பாணம்
"நாவலர்" அச்சுக்கூடம்.
1910.
P5282

Fig. 4 An example of bounding box segmentation.

D. Results and Discussion

The following figures illustrate the character and word accuracy of the Google Cloud Vision API, Google Docs API, Open-source Tesseract OCR, Surya OCR, and OCR Tamil across diverse page formats, document conditions, printing technologies, and processing methods. These comparisons underscore the strengths and limitations of both the Commercial and open-source OCR systems.

The evaluation of optical character recognition (OCR) systems, as shown in Figs. 5 and 6, and detailed in Tables III and IV, reveals significant variations in the performance of different models and page formats. Commercial OCR solutions, such as the Google Cloud Vision API and Google Docs API, consistently outperform open-source alternatives, such as Tesseract OCR, Surya OCR, and OCR Tamil. The analysis demonstrated that commercial models deliver higher accuracy in both character-level and word-level recognition, showcasing their superior performance in handling diverse input types. Among open-source OCR systems, OCR Tamil demonstrates a strong performance across the most diverse page formats compared to other open-source OCR systems. Open-source Tesseract OCR excels in recognizing text in two-column layouts, outperforming other open-source alternatives. In contrast, Surya OCR demonstrates the weakest performance, particularly in handling text with images, oriented text, and advertisements. Notably, in some cases, the OCR systems' accuracy scores at both character and word levels were negative, falling below -100% (e.g., -125.37, -163.01). This was likely due to the OCR output containing significantly more incorrect text than the ground truth. This was typically caused by hallucinated characters, formatting artifacts, or segmentation errors.

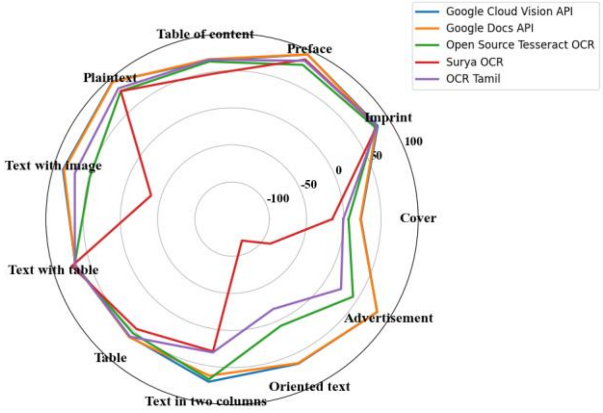


Fig. 5 Character Accuracy for Diverse Page Formats.

TABLE III
CHARACTER ACCURACY (%) OF OCR SYSTEMS ACROSS DIVERSE PAGE FORMATS

Page Format	GCV API	GD API	OST OCR	Surya OCR	OCR Tamil
Cover	22.98	22.35	6.44	-15.45	-0.73
Imprint	82.44	80.65	78.57	80.88	81.17
Preface	93.89	94.03	78.52	86.44	84.58
Table of Content	65.97	67.69	64.54	47.46	67.18
Plain Text	95.28	94.72	78.30	78.13	83.17
Text with Image	86.70	85.20	49.35	-36.75	70.13
Text with Table	69.19	69.41	70.62	74.20	70.07
Table	59.41	60.63	53.40	45.94	59.51
Text in two columns	71.35	63.18	67.96	29.96	31.52
Oriented Text	64.07	63.27	7.93	-118.22	-16.92
Advertisement	81.57	81.78	43.14	-88.92	23.99

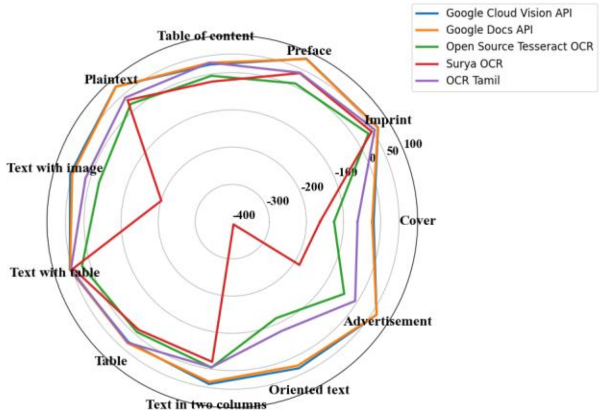


Fig. 6 Word Accuracy for Diverse Page Formats.

TABLE IV
WORD ACCURACY (%) OF OCR SYSTEMS ACROSS DIVERSE PAGE FORMATS

Page Format	GCV API	GD API	OST OCR	Surya OCR	OCR Tamil
Cover	-18.73	-22.80	-125.37	-163.01	-61.96
Imprint	67.47	66.56	36.68	46.90	56.73
Preface	81.08	79.97	8.16	38.40	39.77
Table of Content	26.13	31.01	-4.33	-20.01	31.30
Plain Text	79.72	78.89	18.28	31.51	40.56
Text with Image	55.63	50.12	-25.46	-200.54	12.81
Text with Table	56.89	55.83	22.52	47.26	53.75
Table	27.89	31.55	-7.52	-15.10	30.06
Text in two columns	40.73	34.61	-4.33	-19.47	-4.77
Oriented Text	32.88	25.08	-115.08	-391.90	-77.59
Advertisement	62.62	62.70	-41.11	-184.95	-6.02

Error Analysis of Systems' Performance Across Diverse Page Formats

All systems faced challenges when processing cover pages due to the presence of complex layouts and graphical elements, which pose significant difficulties for OCR systems across the board. Common issues included the generation of extraneous symbols, such as \, &, ", and _ which were absent in the ground truth but appeared owing to graphical artifacts or segmentation errors. Misrecognition of Tamil characters frequently resulted in distorted or incomplete words, with unrelated symbols or incorrect substitutions introduced. For instance, the word குற்றியலிகரம் appeared as குறிறியலிகரம், where characters and vowel modifiers were misinterpreted or omitted entirely, resulting in nonsensical outputs. Tamil's complex ligatures and vowel modifiers were often poorly handled, breaking the word structure (e.g., மூழ்ச்சயும் instead of மகிழ்ச்சியும்). Structural disorganization was another persistent issue, with Tesseract OCR failed to maintain the original text's hierarchy, such as numbered lists or sections (for example ௧௫, ஐஐகாரக்குறுக்கம் instead of ௧௫.ஐகாரக்குறுக்கம். Word formation errors were also common, where characters from unrelated words were joined or fragmented, creating nonsensical phrases (e.g., போர்து.துக்கேய் instead of போர்த்துக்கேயர்). Some errors stemmed from incorrect word segmentation. For example, அதிகாரமூர்த்திகள் was misrecognized as அதிகாரமூர் த்திகள், where an incorrect split introduced a semantic and phonetic disconnect, distorting the intended meaning. The Open-source Tesseract OCR performed reasonably well in recognizing Tamil digit characters. However, other OCR systems failed to recognize Tamil numerals, instead misidentifying them as visually similar Tamil letters, which led to inaccuracies. Surya OCR, in particular, generated additional characters when processing text from oriented pages, significantly reducing its accuracy. Similarly, OCR Tamil processed text continuously in the output file, even when the text appeared in multiple lines in the original image. This caused deviations from the ground truth and introduced additional error rates owing to the formatting inconsistencies. This also affected the overall accuracy of the OCR system. Likewise, Google Docs OCR continuously processed certain portions of text in the output file, particularly when paragraphs or grouped text were present. However, unlike OCR Tamil, this issue occurred only in specific cases rather than affecting the entire text.

The evaluation of OCR systems across varying document conditions, mono/multilingual text, and printing technologies highlights distinct differences in their performance. Figs. 7 and 8, and detailed in Tables V and VI, show the performance of each system across different document conditions, demonstrating that commercial OCR solutions, such as Google Cloud Vision API and Google Docs API, consistently deliver higher accuracy than open-source alternatives such as Tesseract OCR, Surya OCR, and OCR Tamil. These commercial systems demonstrate robustness under all conditions. However, under “noisy condition” or “noisy and damaged condition”, their performance noticeably decreases, although they still outperform the open-source OCR systems. OCR Tamil demonstrates strong word-level accuracy across various conditions, outperforming Surya OCR and Tesseract OCR. Tesseract OCR struggles more with “multilingual text” and “noisy and damaged condition” compared to other OCR systems. Furthermore, documents produced using modern digital print technology yield higher accuracy across all models than older letterpress technologies, highlighting the impact of print quality on OCR performance. Surya OCR generally performs poorly across most conditions. However, it shows relatively better performance in categories such as “multilingual text”, “digital print technology”, “good condition”, and “noisy and damaged condition” when compared to its performance in other areas. Despite this, it still performs worse than the other OCR systems.

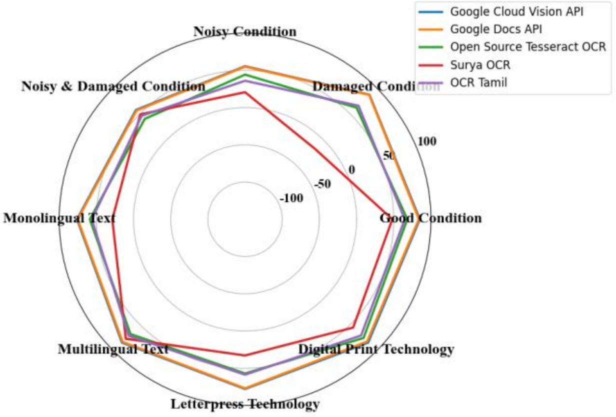


Fig. 7 Character Accuracy for Special Categorization.

TABLE V
CHARACTER ACCURACY (%) OF OCR SYSTEMS FOR SPECIAL CATEGORIZATION

Special Categories	GCV API	GD API	OST OCR	Surya OCR	OCR Tamil
Good condition	85.03	83.85	68.28	48.12	64.87
Damaged condition	86.64	86.14	62.01	-16.58	65.75
Noisy condition	55.58	54.93	44.30	20.67	36.05
Noisy & damaged condition	57.61	56.46	40.72	49.29	46.54
Monolingual text	76.05	75.09	58.79	28.57	56.12
Multilingual text	84.14	82.84	67.94	76.86	71.31
Letterpress technology	77.84	76.98	57.06	32.85	58.67
Digital print technology	84.11	82.16	75.37	55.61	70.67

TABLE VI
WORD ACCURACY (%) OF OCR SYSTEMS FOR SPECIAL CATEGORIZATION

Special Categories	GCV API	GD API	OST OCR	Surya OCR	OCR Tamil
Good condition	68.63	67.79	8.96	-17.90	29.22
Damaged condition	66.81	66.67	-6.11	-79.59	30.59
Noisy condition	27.62	24.37	-55.27	-76.65	-8.61
Noisy & damaged condition	19.13	15.17	-62.52	-46.69	-15.58
Monolingual text	53.78	52.11	-15.61	-51.71	14.00
Multilingual text	65.96	65.24	7.20	36.47	38.72
Letterpress technology	53.70	52.02	-20.10	-49.78	11.76
Digital print technology	71.29	70.49	23.27	14.65	50.98

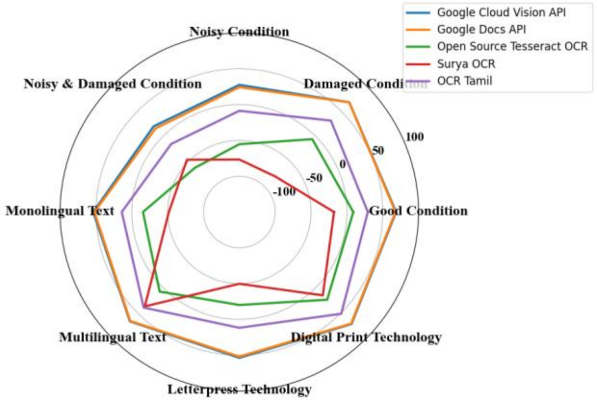


Fig. 8 Word Accuracy for Special Categorization.

Error Analysis - For special categorization

One factor that affected the performance of all the systems under noisy conditions was that the ground truth did not include labels or rubber stamps, even though the systems recognized them. This mismatch likely led to a lower accuracy, as the systems detected these extra elements, but they were not in the reference data. For instance, elements like “UNIVERSITY OF JAFFNA”, “ARCHIVES”, “Digitized by Noolaham Foundation”, and “noolaham.org”, which appeared on rubber stamps and stickers, were incorrectly included as part of the recognized text. A major source of error was the system’s difficulty in handling text layout and segmenting text from non-textual components such as logos, stamps, or graphical elements. Surya OCR, in particular, struggled with this aspect, leading to the inclusion of extraneous symbols and irrelevant text. Another common issue was the confusion caused by bilingual text, particularly under conditions of low character clarity. In such cases, English characters were often misinterpreted as Tamil. For instance, the phrase “KAILASAPATHIYUM NAANUM” was misrecognized as “ப \KAILASAPATHIYUM NAANUM”, “A CRITICAL ASSESSMENT” appeared as “பல்க் CRITICAL ASSESSMENT”, and “வெ. சாமிநாத சர்மா” appeared as “Qal. சாமி நாத சர்மா”. Additionally, the system faced challenges in segmentation, such as fragmenting “திருவாளர்-திரு.வி.கலியாணசுந்தரனார்” into “இருவாளர் - திரு. வி. கலியாணசுந்தரனார்” breaking the natural word flow. These segmentation issues, coupled with the system's inability to handle Tamil's complex ligatures and vowel modifiers effectively, further compounded the errors at the word level, leading to distorted or nonsensical outputs.

The evaluation results revealed distinct performance patterns among the five OCR systems across the various processing methods, as shown in Figs. 9 and 10. Commercial OCR systems, particularly Google Cloud Vision API and Google Docs API, consistently deliver higher accuracy in both character-level and word-level recognition compared with open-source solutions such as Tesseract OCR, Surya OCR, and OCR Tamil. Commercial systems demonstrate notable reliability in both the full-image and line-by-line segmentation approaches. Surya OCR and OCR Tamil showed significant weaknesses in word-level recognition, particularly in line-by-line segmentation.

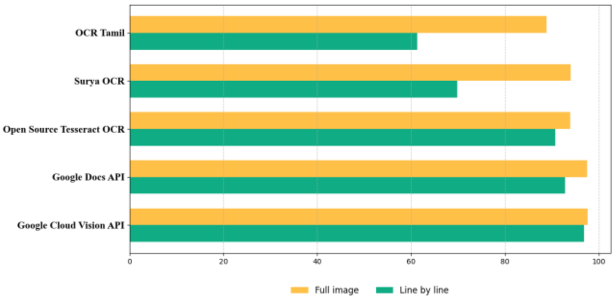


Fig. 9 Comparison of Character Accuracy for Different Methods (Full Image vs. Line-by-Line).

Error Analysis - For Line-by-Line Segmentation

Surya OCR faced significant challenges, including misrecognition of characters, misalignment or omission of vowel modifiers, and improper segmentation of words.

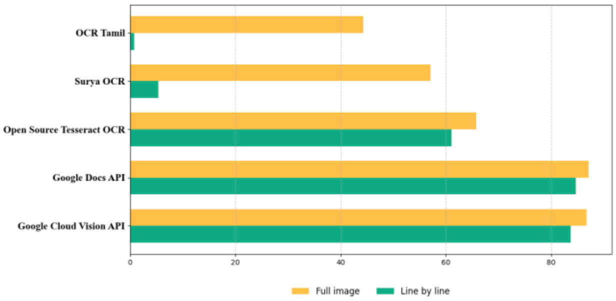


Fig. 10 Comparison of Word Accuracy for Different Methods (Full Image vs. Line-by-Line).

Additionally, random characters such as repeated numerals or nonsensical sequences often appeared in the output, likely caused by background noise or artifacts in the input image. The system also struggled to differentiate between meaningful text and non-text regions, frequently hallucinating text in areas where no text existed. OCR Tamil, on the other hand, performed well when processing full images; however, when handling cropped images line-by-line, it generated output text without proper spacing between words. This lack of spacing led to lower word-level accuracy, although the character-level accuracy remained promising. Similarly, Tesseract OCR exhibited several notable weaknesses in line-by-line segmentation. A prominent issue was the frequent hallucination of characters, particularly in the incorrect rendering of Tamil numerals, punctuation, and conjunct consonants. In many cases, it inserted irrelevant symbols or random Latin script tokens (e.g., “DLO”, “IVS HO”, and “Amy br”), indicating a failure to filter out background noise or non-textual artifacts, especially in complex poetic or dense prose segments. Tesseract also suffered from alignment drift in longer lines, resulting in misplaced character stacking and visual clutter.

The results demonstrated that the performance of OCR systems is significantly influenced by the use of bounding-box segmentation techniques. In terms of Character Accuracy (CA) and Word Accuracy (WA), both the Google Cloud Vision API and Google Docs API show notable improvements when processing cropped images compared to the original unprocessed images. This suggests that isolating text regions through bounding boxes effectively enhances recognition accuracy by reducing noise and irrelevant visual elements in the input, as shown in Figs. 11 and 12, respectively.

In contrast, open-source systems, such as Tesseract OCR, Surya OCR, and OCR Tamil, struggle with cropped images and exhibit significantly lower performance. Surya OCR, in particular, experienced a dramatic decline in CA and WA, performing far worse on cropped images than on the original images. OCR Tamil, while moderate in its performance on original images, demonstrates a relatively better accuracy than Surya OCR when handling cropped images, although its performance remains below that of commercial solutions. These results highlight the limitations of open-source systems in managing segmented inputs, emphasizing the need for improved segmentation techniques and contextual post-processing to enhance their effectiveness.

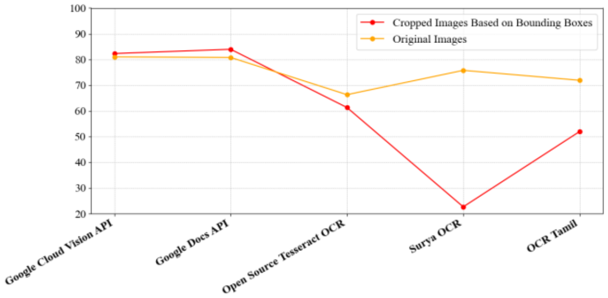


Fig. 11 Comparison of Character Accuracy for Different Methods (Cropped Images Based on Bounding Boxes vs. Original Image).

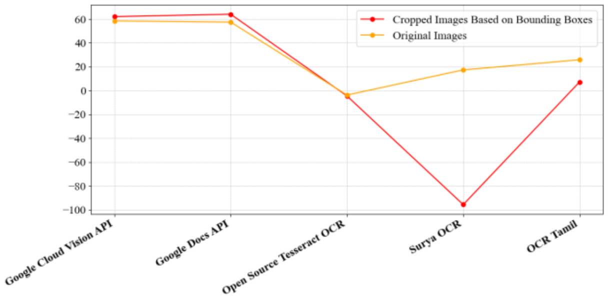


Fig. 12 Comparison of Word Accuracy for Different Methods (Cropped Images Based on Bounding Boxes vs. Original Image).

Error Analysis - For bounding-box segmentation

One possible explanation for the decrease in the performance of open-source OCR systems for cropped images based on bounding boxes compared to the original image is the reduction in image quality during the segmentation process. Cropping often led to decreased resolution, particularly for smaller text regions, and introduced distortions or noise that hindered accurate text recognition. Additionally, the contextual information present in the original image, such as neighboring text or layout structure, was often lost during segmentation, further impacting the OCR system's ability to interpret content accurately. In particular, the Surya OCR revealed significant challenges at the global level. The system struggled to accurately recognize and reproduce Tamil texts, particularly when dealing with complex layouts or mixed graphical elements. Key issues included the introduction of extraneous symbols (e.g., “ஆக்கியோர்” misrepresented as “ஆக்கியோர் . .”, incomplete or fragmented text recognition, unnecessary line breaks, and spacing errors. For instance, the phrase “தக்ஷிணைகைலாச புராணம்.” was fragmented as “தத்து ணைகலாச புராணம்.”, thus distorting its original meaning. Additionally, some bounding boxes contained entirely unrelated text, such as “Camera Brannel 31

TABLE VII
EXAMPLES OF OCR ERROR TYPES IN TAMIL TEXT

Error type	OCR output	Ground truth
Vowel modifier misrecognition	கீரிமலேத்	கீரிமலைத்
Extra symbols/artifacts	ஜஐகாரக்குறுக்கம்	ஐகாரக்குறுக்கம்
Tamil numeral misread	கட	கஉ
Word split/merge errors	நாட்டிலி ருந்து	நாட்டிலிருந்து
Punctuation misrecognition	கடு, எழுச்சு	கரு. எழுத்து
Language switching errors	கே. எஸ். சிவகுமாரன் ந.க.	கே.எஸ்.சிவகுமாரன் B.A.
Character substitution	வல்லெடுத்து	வல்லெழுத்து
True word error	அணு	அணா
Line structure loss	யாழ்ப்பாணத்து ப்பு வறதமிழ்ச்சங்கத்தா ருடைய விருப்பத்தின்படி	யாழ்ப்பாணத்து தமிழ்ச்சங்கத்தாருடைய விருப்பத்தின் படி
Redundant glyphs	உதாரணஙவ்களே	உதாரணங்களே

Concession : CATHERSENTIAL PRODUCE”, which was erroneously introduced and irrelevant to the original content. The system also struggled with Tamil-specific features such as vowel modifiers, ligatures, and symbols, often misinterpreting or entirely omitting them.

We have collected all the errors from the evaluations, analyzed them and found many of the errors fall into ten types. Table VII presents representative examples of the common OCR error types identified during the evaluation. By analyzing these error types, it becomes possible to identify where targeted post-processing approaches may help reduce OCR errors in future systems.

IV. CONCLUSION

Tamil Optical Character Recognition (OCR) has seen notable advancements in recent years, with the use of deep learning techniques and post-processing methods, including large language models, to improve system accuracy. However, challenges remain in handling complex layouts, degraded documents, historical Tamil symbols, and handwritten texts. The limited availability of large and diverse datasets and standardized benchmarks has also slowed the progress of OCR development.

We developed a diverse dataset of 164 scanned images from the Library, University of Jaffna, and the Noolaham Foundation, covering publications from 1850 onward. The dataset includes a wide range of document types and is categorized by attributes such as image condition, language composition, and printing technology. Ground-truth text was carefully prepared through a blend of automated extraction and expert manual correction, with bounding box annotations to support detailed evaluation. This comprehensive dataset enables a realistic and thorough assessment of OCR system performance and serves as a valuable benchmark for advancing Tamil OCR research.

The evaluation of commercial OCR systems, such as Google Cloud Vision API and Google Docs API, showed higher character- and word-level accuracy compared to open-source systems such as Tesseract OCR, Surya OCR, and OCR Tamil, especially for well-maintained and digitally printed documents. However, all the systems showed reduced performance with cover pages, noisy documents, and damaged text, reflecting the impact of document quality on the OCR results.

A small performance difference was observed between line-by-line and full-image processing. Full-image processing performs slightly better because of its ability to capture the contextual relationships across documents. This suggests that post-processing could further enhance the accuracy. Based on our observations, full-image processing may allow the OCR systems to make more consistent post-processing steps when the entire page is available, although the specific internal mechanisms used by these systems are not publicly documented. In the full-image mode, OCR systems may capture contextual information based on the surrounding text, and the systems may correct errors using this context. Nonetheless, the choice of the method may depend on document-specific factors.

Bounding box segmentation improved recognition accuracy for commercial systems, whereas open-source systems, particularly Surya OCR, encountered challenges, likely due to the lower image quality resulting from the segmentation process.

We also analyzed the OCR outputs and categorized the errors into ten types. These error patterns indicate where future improvements could be directed. For example, handling of vowel modifiers could be strengthened, discrimination of visually similar characters could be improved, and robustness against degradation in historical documents could be enhanced. In addition, more reliable word-boundary handling would help address word split or merge issues. Different post-processing approaches could be explored in future work to address these categories and further improve OCR accuracy.

As discussed in this paper, Tamil text forms a small portion of large language model datasets. Despite Tamil's long literary tradition and global speaker base exceeding 80 million, many documents, particularly manuscripts on ola leaves, remain undigitized.

Future research should focus on creating more adaptable OCR systems capable of handling diverse inputs. Integrating large language models into post-processing workflows can help close the gap between the OCR outputs and meaningful text interpretation. Tamil OCR systems have the potential to not only transcribe text but also support information retrieval, digital preservation, and knowledge extraction from Tamil literature and historical archives.

Acknowledgment

This research study was carried out under the DigSAL project, supported by the German Academic Exchange Service (DAAD) and funded by the Federal Ministry for Economic Cooperation and Development (BMZ), Germany through SDG Partnerships (2024-2027).

References

[1] C.-H. Liu, A. Karakanta, A. Tong, O. Aulov, I. Soboroff, J. Washington, and X. Zhao, "Introduction to the Special Issue on Machine Translation for Low-Resource Languages," *Mach. Transl.*, vol. 34, Jan. 2021.

[2] P. Bhattacharyya, H. Murthy, S. Ranathunga, and R. Munasinghe, "Indic language computing," *Commun. ACM*, vol. 62, no. 11, pp. 70–75, 2019. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/3343456

[3] K. Sarveswaran, G. Dias, and M. Butt, "ThamizhiMorph: A morphological parser for the Tamil language," *Mach. Transl.*, vol. 35, no. 1, pp. 37–70, Apr. 2021. [Online]. Available: https://doi.org/10.1007/s10590-021-09261-5

[4] A. S. and H. G, "Recognition of Historical Records Using Gabor and Zonal Features," *Signal Image Processing : An International Journal*, vol. 6, pp. 57–69, Aug. 2015.

[5] K. Dutta, "Handwritten Word Recognition for Indic & Latin Scripts Using Deep CNN-RNN Hybrid Networks," Ph.D. dissertation, IIIT Hyderabad, 2019, accessed: Dec. 4, 2024.

[6] R. Krithiga, S. R. Varsini, R. G. Joshua, and C. U. Om Kumar, "Ancient Character Recognition: A Comprehensive Review," *IEEE Access*, vol. 13, pp. 88 847–88 857, 2025.

[7] N. Pradeep, D. Subramanian, and M. Ganapathy, "Digitizing India's Ancient Texts: AI for Tamil Palm Leaf Manuscript Preservation and Accessibility," [Online]. Available: https://www.academia.edu/download/120320058/NANDHINI_PRADEEPv1.pdf, 2024, accessed: Dec.12, 2024.

[8] R. Arjunan, R. S. Shankar, M. Asuti, N. Benni, N. Siddappa, P. Challagidad, and V. Bhandage, "Deciphering Ancient Tamil Epigraphy: A Deep Learning Approach for Vatteluttu Script Recognition," *Journal of Internet Services and Information Security*, vol. 15, pp. 451–467, Feb. 2025.

[9] S. Rajendran, M. Anand Kumar, R. Rajalakshmi, V. Dhanalakshmi, P. Balasubramanian, and K. P. Soman, "Tamil NLP Technologies: Challenges, State of the Art, Trends and Future Scope," in *Speech and Language Technologies for Low-Resource Languages*, M. Anand Kumar, B. R. Chakravarthi, B. Bharathi, C. O'Riordan, H. Murthy, T. Durairaj, and T. Mandl, Eds. Cham: Springer International Publishing, 2023, pp. 73–98.

[10] R. Jayadevan, S. R. Kolhe, P. M. Patil, and U. Pal, "Offline 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, 2011.

[11] B. Murugan and P. Visalakshi, "Ancient Tamil inscription recognition using detect, recognize and labelling, interpreter framework of text method," *Heritage Science*, vol. 12, no. 1, p. Article 122, 2024. [Online]. Available: https://doi.org/10.1186/s40494-024-01522-9

[12] K. G. Aparna and A. G. Ramakrishnan, "A complete tamil optical character recognition system," in *Proceedings of the 5th International Workshop on Document Analysis Systems V*, ser. DAS '02. Berlin, Heidelberg: Springer-Verlag, 2002, p. 53–57.

[13] V. Jayanthi and S. Thenmalar, "A review on recognizing offline Tamil manuscript character," in AIP Conference Proceedings, vol. 2591, no. 1. AIP Publishing, 2023.

[14] N. Shaffi and F. Hajamohideen, "uTHCD: A New Benchmarking for Tamil Handwritten OCR," *IEEE Access*, vol. 9, Jul. 2021.

[15] FutureBeeAI, "Tamil Language Printed OCR Image Datasets," [Online]. Available: https://www.futurebeeai.com/dataset/printed-ocr-image-data-sets/tamil, 2023, accessed: Sep. 2, 2025.

[16] I. R. Work, "Tamil OCR Object Detection Dataset," [Online]. Available: https://universe.roboflow.com/iitbresearchwork/tamil-ocr-z1rsy, 2023, accessed: Jun. 28, 2025.

[17] DataoceanAI, "Tamil OCR Image Corpus," [Online]. Available: https://dataoceanai.com/datasets/ocr/tamil-ocr-image-dataset/, 2023, accessed: Jun. 7, 2025.

[18] G. L. Hart, "Statement on Tamil as a Classical Language," [Online]. Available: https://southasia.berkeley.edu/language/tamilberkeley/statement-tamil-classical-language, 2000, accessed: Mar. 21, 2024.

[19] A. Balachandran, "Tamil-Llama: A New Tamil Language Model Based on Llama 2," 2023. [Online]. Available: https://arxiv.org/abs/2311.05845

[20] Analytics India Magazine, "How Good is LLaMA 3 for Indic Languages?" [Online]. Available: https://analyticsindiamag.com/ai-origins-evolution/how-good-is-llama-3-for-indic-languages/, 2024, accessed: Apr. 15, 2025.

[21] W. Q. Leong, J. G. Ngui, Y. Susanto, H. Rengarajan, K. Sarveswaran, and W. C. Tjhi, "BHASA: A Holistic Southeast Asian Linguistic and Cultural Evaluation Suite for Large Language Models," 2023. [Online]. Available: https://arxiv.org/abs/2309.06085

[22] M. A. R. Raj, S. Abirami, S. M. Shyni, and S. Murugappan, "Handwritten Tamil OCR by Using the Statistical and Structural Theory," [Online]. Available: https://uttamam.org/papers/19_18.pdf, 2019, accessed: Dec. 14, 2024.

[23] A. R. R. M., A. S., and M. S., "Analysis of Statistical Feature Extraction Approaches Used in Tamil Handwritten OCR," [Online]. Available: https://uttamam.org/papers/13_32_.pdf, 2013, accessed: Dec. 16, 2024.

[24] N. D. L. Sundaram, "Embedding Co-Features in 'OCR-Friendly' Fonts will go a Longway in Machine Reading of Texts," [Online]. Available: https://uttamam.org/papers/10_62.pdf, 2010, accessed: Dec. 2, 2024.

[25] Unicode Consortium, "Tamil," [Online]. Available: https://www.unicode.org/charts/PDF/U0B80.pdf, accessed: Apr. 22, 2024.

[26] Unicode Consortium, "Tamil Supplement," [Online]. Available: https://www.unicode.org/charts/PDF/U11FC0.pdf, accessed: Apr. 22, 2024.

[27] C. Liyanage, T. Nadungodage, and R. Weerasinghe, "Developing a commercial grade Tamil OCR for recognizing font and size independent text," in *2015 Fifteenth International Conference on Advances in ICT for Emerging Regions (ICTer)*. IEEE, 2015, pp. 130–134.

[28] C. Vasantharajan, L. Tharmalingam, and U. Thayasivam, "Adapting the Tesseract Open-Source OCR Engine for Tamil and Sinhala Legacy Fonts and Creating a Parallel Corpus for Tamil-Sinhala-English," in *2022 International Conference on Asian Language Processing (IALP)*, 2022, pp. 143–149.

[29] M. Monisha and V. F. Enigo, "Complexities in Developing Tamil-Brahmi Script OCR: An Analysis," in *19th Tamil Internet Conference*, 2020, pp. 88–101.

[30] J. K. Raju and R. Prabhakar, "A Comparative Study of Optical Character Recognition for Tamil Script," *European Journal of Scientific Research*, vol. 35, Jan. 2009.

[31] M. A. Hossain and S. Afrin, "Optical Character Recognition based on Template Matching," *Global Journal of Computer Science and Technology*, vol. 19, pp. 31–35, May. 2019.

[32] A. Murugappan, V. Essakiammal, and B. Ramachandran, "Statistical features based character recognition for offline handwritten Tamil document images using HMM," *International Journal of Computational Vision and Robotics*, vol. 5, p. 422, Oct. 2015.

[33] M. A. R. Raj and S. Abirami, "Structural representation-based off-line Tamil handwritten character recognition," Soft Comput., vol. 24, no. 3, pp. 1447–1472, 2019.

[34] L. Yang, Z. Wu, T. Xu, J. Du, and E. Wu, "Easy recognition of artistic Chinese calligraphic characters," *Vis. Comput.*, vol. 39, no. 8, pp. 3755–3766, 2023. [Online]. Available: https://doi.org/10.1007/s00371-023-03026-2

[35] A. G. Zuniga, J. B. Florindo, and O. M. Bruno, "Gabor wavelets combined with volumetric fractal dimension applied to texture analysis," *Pattern Recognit. Lett.*, vol. 36, pp. 135–143, 2014.

[36] G. S. Eskander, R. Sabourin, and E. Granger, "Dissimilarity representation for handwritten signature verification," *CEUR Workshop Proc.*, 2013.

[37] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[38] R. Kumar, A. Singh, and R. Sharma, "Optical Character Recognition Using Modified Gabor Filter," *International Journal of Advance Research in Science and Engineering*, vol. 4, no. 1, pp. 123–130, 2015.

[39] U. Pal and B. Chaudhuri, "Indian script character recognition: a survey," *Pattern Recognit.*, vol. 37, no. 9, pp. 1887–1899, 2004.

[40] S. Krishnamoorthy, R. Loganathan, S. CJ, A. Vadakkepatt, and S. Kp, "Multiclass Hierarchical SVM for Recognition of Printed Tamil Characters," *Proceedings of IJCAI 2007 Workshop on Analytics for Noisy Unstructured Text Data, AND 2007*, Jan. 2007.

[41] M. Shafana, R. Ragel, and T. Kumara, “An effective feature set for enhancing printed Tamil character recognition,” *Journal of the National Science Foundation of Sri Lanka*, Sep. 2021.

[42] T. Patnaik, S. Gupta, C. jawahar, S. Chaudhury, and A. Ramakrishnan, “Design and Evaluation of Omnifont Tamil OCR,” Jun. 2010.

[43] K. Shashikiran, K. S. Prasad, R. Kunwar, and A. G. Ramakrishnan, “Comparison of HMM and SDTW for Tamil handwritten character recognition,” in *2010 International Conference on Signal Processing and Communications (SPCOM)*, 2010, pp. 1–4.

[44] N. H. Barnouti, M. Abomaali, and M. H. N. Al-Mayyahi, “An efficient character recognition technique using K-nearest neighbor classifier,” *International Journal of Engineering and Technology*, vol. 7, no. 4, pp. 3148–3153, 2018.

[45] S. Tian, S. Lu, B. Su, and C. L. Tan, “Scene Text Recognition Using Co-occurrence of Histogram of Oriented Gradients,” in *2013 12th International Conference on Document Analysis and Recognition*, 2013, pp. 912–916.

[46] L. Wang, W. Zhang, and M. Liu, “Handwritten English Character Recognition Using SIFT,” *Int. J. Pattern Recognit Artif Intell.*, vol. 26, no. 8, p. 1250012, 2012.

[47] Y. LeCun, Y. Bengio, and G. Hinton, “Deep Learning,” Nature, vol. 521, pp. 436–444, 2015.

[48] J. Schmidhuber, “Deep Learning in Neural Networks: An Overview,” *Neural Networks*, vol. 61, pp. 85–117, 2015.

[49] H. Guo, Y. Liu, D. Yang, and J. Zhao, “Offline handwritten Tai Le character recognition using ensemble deep learning,” *Vis. Comput.*, vol. 38, no. 11, pp. 3897–3910, Nov. 2022. [Online]. Available: https://doi.org/10.1007/s00371-021-02230-2

[50] I. Goodfellow, Y. Bengio, and A. Courville, “Deep Learning,” *MIT Press*, 2016.

[51] T. Pappacena, “How to Use Deep Learning-Based OCR: A Technical Deep-Dive Into Implementation,” [Online]. Available: https://landing.ai/blog/how-to-use-deep-learning-based-ocr-a-technical-deep-dive-into-implementation, 2024, accessed: Jun. 8, 2025.

[52] O. Ignat, J. Maillard, V. Chaudhary, and F. Guzm´an, “OCR improves machine translation for low-resource languages,” 2022. [Online]. Available: https://arxiv.org/abs/2202.13274

[53] A. T. Maung, S. Salekin, and M. A. Haque, “A hybrid approach to Bangla handwritten OCR: combining YOLO and an advanced CNN,” *Discover Artificial Intelligence*, vol. 5, no. 1, p. 119, 2025. [Online]. Available: https://doi.org/10.1007/s44163-025-00251-7

[54] A. Naseer and K. Zafar, “Meta features-based scale invariant OCR decision making using LSTM-RNN,” *Comput. Math. Organ. Theory*, vol. 25, Jun. 2019.

[55] I. Anuradha, C. Liyanage, and R. Weerasinghe, “Estimating the Effects of Text Genre, Image Resolution and Algorithmic Complexity needed for Sinhala Optical Character Recognition,” *International Journal on Advances in ICT for Emerging Regions (ICTer)*, vol. 14, p. 43, Aug. 2021.

[56] N. Gupta and A. S. Jalal, “Traditional to transfer learning progression on scene text detection and recognition: a survey,” *Artif. Intell. Rev.*, vol. 55, no. 4, pp. 3457–3502, 2022.

[57] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS’17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6000–6010.

[58] Y. Sun, X. Xie, Z. Li, and K. Yang, “Batch-transformer for scene text image super-resolution,” *Vis. Comput.*, vol. 40, no. 10, pp. 7399–7409, Oct. 2024. [Online]. Available: https://doi.org/10.1007/s00371-024-03598-7

[59] J. Castro, S. Arauco C, C. Villalobos, F. Cordeiro, A. Alexandre, and M. Pacheco, “Improvement Optical Character Recognition for Structured Documents using Generative Adversarial Networks,” Sep. 2021, pp. 285–292.

[60] R. Smith, “An Overview of the Tesseract OCR Engine,” in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, vol. 2, 2007, pp. 629–633.

[61] T. Hegghammer, “OCR with Tesseract, Amazon Textract, and Google Document AI: a benchmarking experiment,” *J. Comput. Social Sci.*, vol. 5, pp. 861–882, 2022.

[62] P. M. Vibhute and M. S. Deshpande, “Optical Character Recognition (OCR) of Marathi Printed Documents Using Statistical Approach,” in *Advances in Computing and Data Sciences: Second International Conference, ICACDS 2018, Dehradun, India, April 20-21, 2018, Revised Selected Papers, Part I 2*. Springer, 2018, pp. 489–498.

[63] K. Wang, J. Jin, and Q. Wang, “High Performance Chinese/English Mixed OCR with Character Level Language Identification,” in *2009 10th International Conference on Document Analysis and Recognition*, 2009, pp. 406–410.

[64] W. Zhu, N. Sokhandan, G. Yang, S. Martin, and S. Sathyanarayana, “DocBed: A Multi-Stage OCR Solution for Documents with Complex Layouts,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 11, 2022, pp. 12 643–12 649.

[65] D. Gnana Prasath, “Tamil OCR,” [Online]. Available: https://github.com/gnana70/tamil_ocr, Jan. 2024, accessed: Oct. 12, 2024.

[66] M. Ramanan, A. Ramanan, and E. Charles, “A hybrid decision tree for printed Tamil character recognition using SVMs,” in *2015 Fifteenth International Conference on Advances in ICT for Emerging Regions (ICTer)*, 2015, pp. 176–181.

[67] A. Ramakrishnan and K. Mahata, “A complete OCR for printed Tamil text,” Jul. 2000, doi: 10.13140/RG.2.1.4593.5209.

[68] A. Kanakatte and A. Ramakrishnan, “Tamil Gnani - an OCR on Windows,” Aug. 2001.

[69] V. Krishnamoorthy, “OCR Software for Printed Tamil Text,” *Tamil Internet*, pp. 99–101, 2002.

[70] K. Aparna and V. Chakravarthy, “A complete OCR system development of Tamil magazine documents,” *Tamil Internet*, pp. 45–51, 2003.

[71] M. Mathew, A. Mondal, and C. Jawahar, “Towards Deployable OCR Models for Indic Languages,” in *International Conference on Pattern Recognition*. Springer, 2025, pp. 167–182.

[72] M. Ramanan, A. Ramanan, and E. Charles, “A preprocessing method for printed Tamil documents: Skew correction and textual classification,” in *2015 IEEE Seventh International Conference on Intelligent Computingand Information Systems (ICICIS)*, 2015, pp. 495–500.

[73] G. J. Rama, A. Ramakrishnan, and D. Gupta, “Parallel Processing in OCR - A Multithreaded approach,” *Procs. Tamil Internet*, pp. 107–110, 2002.

[74] R. Sakuntharaj and S. Mahesan, “A novel hybrid approach to detect and correct spelling in Tamil text,” in *2016 IEEE International Conference on Information and Automation for Sustainability (ICIAfS)*, 2016, pp. 1–6.

[75] S. Murugan, T. A. Bakthavatchalam, and M. Sankarasubbu, “Symspell and LSTM based Spell-Checkers for Tamil,” in *Tamil Internet Conference*, vol. 11, 2020.

[76] A. Aravinthan and C. Eugene, “Exploring Recent NLP Advances for Tamil: Word Vectors and Hybrid Deep Learning Architectures,” *International Journal on Advances in ICT for Emerging Regions (ICTer)*, Oct 2024.

[77] R. Rajalakshmi, V. Sharma, and A. K. M, “Context Sensitive Tamil Language Spellchecker Using RoBERTa,” in *Speech and Language Technologies for Low-Resource Languages*, A. K. M, B. R. Chakravarthi, B. B, C. O’Riordan, H. Murthy, T. Durairaj, and T. Mandl, Eds. Cham: Springer International Publishing, 2023, pp. 51–61.

[78] K. Uthayamoorthy, K. Kanthasamy, T. Senthaalan, K. Sarveswaran, and G. Dias, “DDSpell - A Data Driven Spell Checker and Suggestion Generator for the Tamil Language,” in *2019 19th International Conference on Advances in ICT for Emerging Regions (ICTer)*, vol. 250, 2019, pp. 1–6.

[79] A. Ramakrishnan and B. Urala Kota, “Online handwritten Tamil word recognition using segmentation, bigram models and verification,” Dec. 2012.

[80] K. Kukich, “Techniques for automatically correcting words in text,” *ACM Comput. Surv.*, vol. 24, no. 4, p. 377–439, Dec. 1992. [Online]. Available: https://doi.org/10.1145/146370.146380

[81] R. Sakuntharaj and S. Mahesan, “Detecting and correcting real-word errors in Tamil sentences,” *Ruhuna J. Sci.*, vol. 9, no. 2, p. 150, 2018.

[82] R. Sridhar, L. Rathi, P. Rithya, and P. Nivrutha, “Use of Tamil grammar rules for correcting errors in optical character recognised document,” in *Tamil Internet Conference*, 2013.

[83] T. M. Breuel, “The OCRopus open source OCR system,” in *Document recognition and retrieval XV*, vol. 6815. SPIE, 2008, pp. 120–134.

[84] M.-Y. Hwang, Y. Shi, A. Ramchandani, G. Pang, P. Krishnan, L. Kabela, F. Seide, S. Datta, and J. Liu, “DISGO: Automatic End-to-End Evaluation for Scene Text OCR,” *arXiv preprint arXiv:2308.13173*, Aug. 2023.

[85] S. Ouzerrout, “Universal-WER: Enhancing WER with segmentation and weighted substitution for varied linguistic contexts,” in *Proceedings of the 9th International Workshop on Computational Linguistics for Uralic Languages*, M. Hämäläinen, F. Pirinen, M. Macias, and M. Crespo Avila, Eds. Helsinki, Finland: Association for Computational Linguistics, Nov. 2024, pp. 29–35. [Online]. Available: https://aclanthology.org/2024.iwclul-1.3/

[86] M. Jenckel, S. S. Bukhari, and A. Dengel, “Transcription Free LSTM OCR Model Evaluation,” in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2018, pp. 122–126.

[87] R. Karpinski, D. Lohani, and A. Belaid, “Metrics for Complete Evaluation of OCR Performance,” in *IPCV’18 - The 22nd Int’l Conf on Image Processing, Computer Vision, & Pattern Recognition*, Las Vegas, United States, Jul. 2018. [Online]. Available: https://inria.hal.science/hal-01981731

[88] Library, University of Jaffna, “Library, university of jaffna,” [Online]. Available: https://www.lib.jfn.ac.lk/, 2025, accessed: Mar. 23, 2025.

[89] Noolaham Foundation, “Noolaham foundation,” [Online]. Available: https://www.noolahamfoundation.org/wiki/index.php/Main_Page, 2025, accessed: Mar. 23, 2025.

[90] Tzutalin, “LabelImg,” [Online]. Available: https://github.com/tzutalin/labelImg, 2015, accessed: Oct. 5, 2024.

[91] Google Cloud, “Google Cloud Vision API Documentation,” [Online]. Available: https://cloud.google.com/vision/docs, accessed: Oct. 10, 2024.

[92] Google Developers, “Google Docs API Reference,” [Online]. Available: https://developers.google.com/docs/api/reference/rest, accessed: Oct. 11, 2024.

[93] V. Paruchuri, “Surya OCR,” [Online]. Available: https://github.com/VikParuchuri/surya, accessed: Oct. 12, 2024.