ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1370

Creating Tamil Fonts Using Handwritten Text

Dr. T. Nirmal Raj[1]; T. Ramya[2]

[1]Assistant Professor, Department of Computer Science and Applications, SCSVMV [Deemed to be University], Enathur, Kanchipuram
[2]Student, II-YEAR MCA, Department of Computer Science and Applications, SCSVMV [Deemed to be University], Enathur, Kanchipuram

Publication Date: 2026/05/30

Abstract: With the rapid advancement of digital technologies and the increasing demand for personalized digital content, the need to convert handwritten text into reusable and scalable digital formats has become highly significant. In particular, regional languages like Tamil, which possess rich script structures and cultural importance, require efficient digitization techniques to preserve and promote their usage in modern digital environments. This project proposes a comprehensive system designed to convert handwritten Tamil text into a fully functional digital font. The system allows users to write Tamil text manually on paper, scan or capture the document as an image, and upload it into the application. The uploaded handwritten document undergoes a series of image preprocessing steps, including noise removal, grayscale conversion, binarization, normalization, and segmentation, to enhance the quality and prepare the data for accurate recognition. Following preprocessing, the system employs character recognition techniques to identify individual Tamil characters. Feature extraction methods are applied to capture the unique structural patterns of each handwritten character. These features are then analyzed using machine learning or pattern recognition algorithms to accurately classify and map each character to its corresponding Tamil Unicode representation. Once the characters are recognized, the system proceeds to generate scalable digital glyphs. Each glyph is carefully designed to retain the stylistic characteristics of the user's handwriting, ensuring personalization. These glyphs are then assembled into a complete font file (such as TTF or OTF format), which can be installed and used across various applications. The system also provides an interactive user interface that enables users to preview the generated font in real-time. Users can test the font with sample text, make adjustments if necessary, and download the finalized font for use in documents, graphic design, publishing, and other digital platforms.

How to Cite: Dr. T. Nirmal Raj; T. Ramya (2026) Creating Tamil Fonts Using Handwritten Text. *International Journal of Innovative Science and Research Technology*, 11(5), 2476-2482. https://doi.org/10.38124/ijisrt/26may1370

I. INTRODUCTION

In the modern digital era, the transformation of traditional handwritten content into digital formats has become an essential requirement across various domains such as education, publishing, communication, and archival systems. While digital text offers advantages such as easy storage, editing, sharing, and reproduction, handwritten text continues to hold personal, cultural, and artistic significance. Particularly in regional languages like Tamil, handwritten scripts reflect unique individual styles and cultural identity, making their preservation and digitization highly important.

Tamil is one of the oldest and most widely spoken classical languages in the world, with a rich literary heritage and a complex script structure. Despite the availability of numerous digital Tamil fonts, most of them lack personalization and do not capture the uniqueness of an individual's handwriting. This creates a gap between traditional handwritten expression and modern digital communication. As a result, there is a growing need for systems that can effectively convert handwritten Tamil text into personalized digital fonts.

This project focuses on developing a system for the recognition and digitization of handwritten Tamil documents, with the added capability of generating custom Tamil fonts. The system aims to allow users to write Tamil text on paper, upload the handwritten document, and automatically convert it into a usable digital font that retains the user's writing style. This not only enhances user creativity but also provides a meaningful way to preserve personal handwriting digitally.

- *Problem Statement*

The rapid growth of digital communication and content creation has increased the need for converting handwritten documents into digital formats. Although many Optical Character Recognition (OCR) systems are available for recognizing printed and handwritten text, most existing solutions are mainly designed for global languages such as English and provide limited support for regional languages like Tamil. Tamil script contains complex character structures, curves, and combinations, making handwritten character recognition a challenging task.

In the current digital environment, users can access numerous pre-designed Tamil fonts; however, these fonts do not preserve the uniqueness of an individual's handwriting

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1370

style. Existing systems mainly focus on converting handwritten text into editable digital text and lack the capability to generate personalized font files from handwritten Tamil characters. As a result, users who wish to use their own handwriting in digital documents, creative designs, or educational materials often depend on manual font creation methods, which are time-consuming and require technical expertise.

➢ *Existing System Drawbacks*

- *Lack of Personalization*

Existing systems do not support the creation of custom fonts based on individual handwriting styles.

- *Limited Support for Tamil Script*

Many OCR tools are optimized for English and other widely used languages, leading to lower accuracy for Tamil.

- *No Font Generation Capability*

Current systems only convert handwritten text into editable text, not into reusable font files (TTF/OTF).

- *High Dependency on Input Quality*

Poor image quality, noise, or improper lighting can significantly reduce recognition accuracy.

- *Difficulty Handling Variations in Handwriting*

Differences in writing styles, sizes, and spacing make it hard for existing systems to consistently recognize characters.

- *Time-Consuming Manual Alternatives*

Creating fonts manually using design tools requires significant effort, skill, and time.

- *Limited User Interaction*

Most systems do not provide user-friendly interfaces for previewing or refining outputs.

- *Inadequate Preservation of Handwriting Style*

Even when text is recognized, the original style of handwriting is lost in standard font conversion.

➢ *Proposed System*

The proposed system aims to develop an advanced and user-friendly solution for converting handwritten Tamil text into a personalized digital font while preserving the uniqueness of an individual's handwriting style. Unlike existing systems that focus only on text recognition, this system integrates both handwritten character recognition and font generation into a single platform, providing a complete end-to-end solution.

In this system, users are allowed to write Tamil text manually on paper and upload the document as an image through a simple interface. The uploaded image is first processed using various image preprocessing techniques such as noise removal, grayscale conversion, binarization, normalization, and segmentation. These steps enhance the quality of the input and isolate individual characters, ensuring better recognition accuracy.

After preprocessing, the system performs handwritten character recognition by applying feature extraction and pattern recognition techniques. Each Tamil character is analyzed based on its structural features and matched with the corresponding character in the database. The recognized characters are then mapped to standard Tamil Unicode values, ensuring compatibility across different platforms and applications.

➢ *System Architecture*

The system design describes the overall structure, architecture, and workflow of the proposed handwritten Tamil text to font generation system. It explains how different components of the system interact with each other to achieve the desired functionality. The design follows a modular and layered approach to ensure flexibility, scalability, and ease of maintenance.

- *Frontend Layer*

The frontend layer of the handwritten Tamil font generation system is responsible for providing an interactive and user-friendly interface through which users can interact with the application. The frontend is developed using modern web technologies such as HTML, CSS, JavaScript, and Streamlit/React.js to ensure a responsive and visually appealing user experience. This layer allows users to download handwriting templates, upload handwritten Tamil text images, preview uploaded files, generate fonts, and download the final font file. Responsive design techniques are used to support different devices such as desktops, tablets, and mobile phones. The frontend communicates with the backend services through RESTful API calls and displays the processing status, recognition results, and font preview dynamically. Additionally, the interface provides validation mechanisms to ensure that users upload valid image formats such as JPG or PNG for accurate processing.

- *Backend Layer*

The backend layer acts as the core processing unit of the system and manages all business logic and server-side operations. The backend can be developed using Python Flask/Django or Node.js with Express.js. It follows a modular architecture where separate modules handle image preprocessing, segmentation, feature extraction, character recognition, Unicode mapping, and font generation. The backend receives uploaded handwritten images from the frontend and processes them using image processing libraries such as OpenCV and PIL. Machine learning models are integrated into the backend to recognize Tamil handwritten characters accurately. The backend also handles file management, API routing, authentication (if required), and communication with the database. Font generation tools such as FontForge are integrated into the backend to create scalable TrueType (TTF) or OpenType (OTF) font files. Error handling and standardized API responses ensure smooth communication between the frontend and backend layers.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1370

- *Database Layer*

The database layer is responsible for storing and managing user information, uploaded handwritten samples, processed character data, generated font files, and application logs. Databases such as MongoDB or MySQL can be used depending on the project requirements. MongoDB is suitable because it supports flexible document-based storage for images, Unicode mappings, and generated font metadata. The database stores user profiles, uploaded templates, recognized characters, generated glyph information, and downloadable font records. Proper indexing and schema design help improve retrieval speed and system performance. The database layer also supports scalability by allowing multiple users to generate and store personalized fonts simultaneously. Additionally, cloud database integration can be implemented for secure storage and remote access.

- *AI & Media Services*

The AI layer plays a major role in recognizing handwritten Tamil characters and improving the accuracy of the system. Machine learning and deep learning techniques are used to identify handwritten Tamil characters from segmented images. Convolutional Neural Networks (CNNs) are commonly used because they are highly effective in image-based character recognition tasks. TensorFlow, Keras, or Scikit-learn frameworks can be used to train and deploy the recognition models. The AI model analyzes extracted features such as curves, edges, loops, and stroke patterns to classify Tamil characters accurately.

II. METHODOLOGY

The methodology of the handwritten Tamil font generation system describes the step-by-step process used to convert handwritten Tamil text into a personalized digital font. The system combines image processing, machine learning, character recognition, Unicode mapping, and font generation techniques to achieve accurate and scalable font creation. The overall methodology is divided into multiple stages, where the output of one stage becomes the input for the next stage.

➢ *Data Collection and Image Acquisition*

The first step in the methodology is collecting handwritten Tamil character samples from users. Users are provided with a predefined handwriting template containing Tamil characters and character combinations. The user fills the template manually using pen or pencil and uploads the scanned image or captured photograph into the system. The uploaded image is validated to ensure that it is in a supported format such as JPG or PNG and meets the required quality standards.

➢ *Image Preprocessing*

After image acquisition, preprocessing techniques are applied to improve the quality of the handwritten image. Since handwritten documents may contain noise, uneven lighting, shadows, or distortions, preprocessing is essential for improving recognition accuracy. The following preprocessing operations are performed:

- Grayscale Conversion
- Noise Removal
- Image Binarization

➢ *Character Segmentation*

In this stage, the preprocessed handwritten image is divided into smaller components such as lines, words, and individual Tamil characters. Segmentation helps isolate each character accurately for feature extraction and recognition. Techniques such as contour detection, connected component analysis, and projection profiling are used to separate the characters. Proper segmentation ensures that overlapping or connected characters are handled effectively.

➢ *Feature Extraction*

Once segmentation is completed, the system extracts important structural features from each Tamil character. Feature extraction identifies the unique patterns that define a handwritten character. The extracted features include:

- Curves and edges
- Stroke thickness
- Loops and intersections
- Character shape and structure

➢ *Character Recognition Using AI*

The extracted features are passed to the character recognition module, which uses Artificial Intelligence and Machine Learning techniques to identify the handwritten Tamil characters. Convolutional Neural Networks (CNNs) are used because they provide high accuracy in image-based recognition tasks. TensorFlow or Keras frameworks are used to train the recognition model using handwritten Tamil datasets.

➢ *Unicode Mapping*

After character recognition, the identified Tamil characters are mapped to their corresponding Unicode values. Unicode mapping standardizes the recognized characters and ensures compatibility across different operating systems, applications, and digital platforms. This stage converts recognized handwritten symbols into machine-readable Tamil text.

➢ *Glyph Generation*

The recognized characters are transformed into graphical representations known as glyphs. The glyph generation process preserves the original handwriting style of the user while converting the characters into scalable vector-based representations. Each glyph maintains the curves, strokes, and structural appearance of the handwritten character.

➢ *Font Generation*

In this stage, all generated glyphs are compiled into a complete digital font file using FontForge or similar font creation tools. The system creates standard font formats such as:

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1370

- TrueType Font (TTF)
- OpenType Font (OTF)

The font generation module also defines spacing, alignment, kerning, and other typographic properties required for a functional font.

➢ *Font Preview and Download*

After generating the font, the system provides a live preview feature where users can test the generated font using sample Tamil text. Users can verify whether the generated font accurately matches their handwriting style. Once satisfied, users can download the font file and install it on their devices for use in applications such as document editing, graphic design, publishing, and digital communication.

III. SCREENSHOT

➢ *Hand Written Font:*

Fig 1 Hand Written Font

➢ *Template Download Creating Tamil Font*

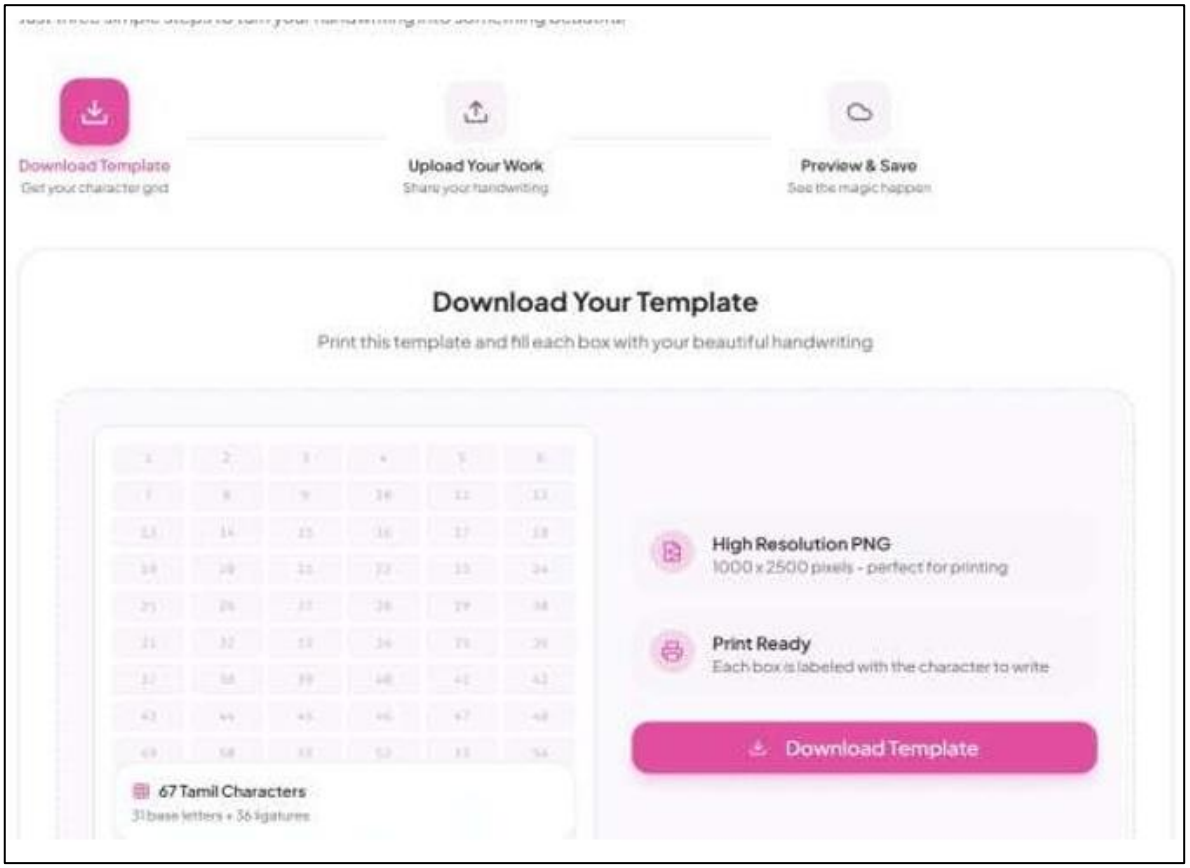

Fig 2 Template Download Creating Tamil Font

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1370

➢ *Technology Stack*

The complete technology stack is summarised below. All dependencies are open-source or available under free tiers appropriate for academic and early-stage commercial use.

Table 1 Technology Stack

Category	Technology	Purpose
Frontend	HTML, CSS, JavaScript, Streamlit / React.js	Used to create an interactive and responsive user interface for uploading handwritten images, previewing fonts, and downloading generated font files.
Backend	Python, Flask / Django	Handles server-side processing, API communication, image processing workflow, and font generation operations.
Image Processing	OpenCV, PIL, NumPy	Used for preprocessing handwritten images, noise removal, segmentation, and image enhancement.
Artificial Intelligence	TensorFlow, Keras, Scikit-learn, CNN	Used for handwritten Tamil character recognition and feature extraction using machine learning and deep learning models.
Font Generation	FontForge, SVG2TTF, FontTools	Generates scalable TTF/OTF font files and manages glyph creation and Unicode mapping.
Database	MongoDB / MySQL	Stores user information, uploaded images, recognized characters, and generated font files.
Cloud Services	MongoDB Atlas, Cloudinary	Provides cloud storage and remote access for uploaded files and generated fonts.
Development Tools	VS Code, PyCharm, GitHub	Used for coding, debugging, version control, and project management.
Platform	Windows 10/11, Chrome Browser	Used as the development and testing environment for the application.

IV. AI-POWERED MODULES

➢ *AI-Powered Modules*

The handwritten Tamil font generation system integrates several AI-powered modules to improve the accuracy, efficiency, and automation of handwritten character recognition and personalized font creation. These modules use machine learning, deep learning, image processing, and pattern recognition techniques to process handwritten Tamil text and convert it into scalable digital fonts while preserving the user's handwriting style.

- *Image Preprocessing Module*

The Image Preprocessing Module uses intelligent image processing techniques to improve the quality of uploaded handwritten images before recognition. Handwritten documents often contain noise, uneven lighting, shadows, and distortions that can affect recognition accuracy.

- *Character Segmentation Module*

The Character Segmentation Module automatically separates handwritten Tamil text into individual characters using AI-based segmentation techniques. Since handwritten Tamil scripts are often connected or irregularly spaced, segmentation becomes a challenging task. This module uses contour detection, connected component analysis, and projection profiling algorithms to accurately isolate each Tamil character.

- *Feature Extraction Module*

The Feature Extraction Module identifies the unique structural properties of handwritten Tamil characters. This module extracts important visual features such as curves, loops, edges, stroke thickness, intersections, and character shapes. The extracted features are converted into numerical representations that can be processed by machine learning models.

- *Handwritten Character Recognition Module*

The Handwritten Character Recognition Module is the core AI component of the system. This module uses deep learning models such as Convolutional Neural Networks (CNNs) to recognize handwritten Tamil characters accurately. The trained AI model analyzes extracted character features and compares them with learned patterns from Tamil handwriting datasets.

- *Unicode Mapping Module*

The Unicode Mapping Module converts recognized Tamil characters into their corresponding Unicode values. AI-supported mapping ensures that the recognized characters are standardized and compatible across different digital platforms and software applications. Unicode mapping acts as a bridge between character recognition and font generation by converting handwritten symbols into machine-readable Tamil text.

- *Glyph Generation Module*

The Glyph Generation Module uses AI-assisted vector generation techniques to create digital glyphs from recognized handwritten characters. This module preserves the original handwriting style, stroke patterns, and structural appearance of the user's writing. The generated glyphs are scalable vector representations that can be resized without losing quality.

- *Font Generation Module*

The Font Generation Module compiles all generated glyphs into a complete digital font file using tools such as FontForge and SVG2TTF. AI-supported alignment and

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1370

spacing techniques help maintain consistent typography, kerning, and character spacing throughout the generated font. The module creates standard font formats such as TrueType Font (TTF) and OpenType Font (OTF), allowing the personalized Tamil font to be installed and used across multiple applications.

- *Font Preview and Validation Module*

The Font Preview and Validation Module provides intelligent preview functionality where users can test the generated font using sample Tamil text. The module validates font readability, Unicode compatibility, and glyph alignment before allowing the font to be downloaded. This AI-assisted validation helps ensure the quality and usability of the generated font.

V. RESULTS AND DISCUSSION

The handwritten Tamil font generation system was successfully designed and implemented to convert handwritten Tamil text into a personalized digital font. The system was tested using multiple handwritten samples collected from different users with varying handwriting styles. The experimental results showed that the system could effectively process handwritten Tamil characters and generate a usable font file that closely preserved the original handwriting style of the user.

➢ *Image Preprocessing*

The preprocessing module successfully improved the quality of handwritten images using grayscale conversion, noise removal, and binarization techniques. This helped increase recognition accuracy.

➢ *Character Segmentation*

The system accurately separated handwritten Tamil text into individual characters, making further processing easier and more efficient.

➢ *Feature Extraction and Recognition*

Important features such as curves, edges, and stroke patterns were extracted successfully. The AI-based CNN model accurately recognized most handwritten Tamil characters.

➢ *Unicode Mapping*

Recognized characters were correctly mapped to Tamil Unicode values, ensuring compatibility across digital platforms and applications.

➢ *Glyph and Font Generation*

The system generated scalable TTF/OTF font files while preserving the user's handwriting style and character appearance.

➢ *Font Preview and Download*

Users were able to preview the generated font using sample Tamil text and download the font for further use.

➢ *Overall System Performance*

The system produced reliable results for most handwritten samples. However, poor image quality and inconsistent handwriting slightly affected recognition accuracy.

VI. CONCLUSION

The handwritten Tamil font generation project presents an innovative approach to bridging the gap between traditional handwriting and modern digital typography. The system successfully integrates image processing, character recognition, and font generation techniques to convert handwritten Tamil text into a personalized digital font. This not only enhances the usability of handwritten content but also preserves the unique writing style of individuals in digital form.

One of the key achievements of this project is the ability to generate custom fonts that reflect personal handwriting, which is not supported by most existing systems. The modular design of the system ensures efficient processing and allows for easy maintenance and future enhancements. By incorporating Unicode mapping, the system ensures that the generated fonts are compatible with various platforms and applications, making them practical for real-world use.

FUTURE SCOPE

The handwritten Tamil font generation system developed in this project provides a strong foundation for converting handwritten text into personalized digital fonts. However, there are several areas where the system can be further improved and extended to enhance its functionality, accuracy, and usability. One of the major enhancements involves improving the accuracy of character recognition by integrating advanced deep learning techniques such as Convolutional Neural Networks (CNNs) and Transformer-based models. These models can better handle variations in handwriting styles, complex character structures, and noisy inputs. Training the system with a larger and more diverse dataset of handwritten Tamil characters will further increase its robustness and performance. Another important enhancement is the expansion of the system to support multiple languages in addition to Tamil. By extending the architecture to handle other regional and international languages, the system can be made more versatile and widely applicable. This would require training models for different scripts and adapting the Unicode mapping and glyph generation modules accordingly. The system can also be enhanced by introducing real-time handwriting recognition capabilities. Instead of uploading images, users could directly write on digital devices such as tablets or touchscreens, and the system would instantly convert the input into a font.

REFERENCES

[1]. Gonzalez, R. C., & Woods, R. E. *Digital Image Processing*, 4th Edition, Pearson Education, 2018.
[2]. Pratt, W. K. *Digital Image Processing: PIKS Scientific Inside*, Wiley-Interscience, 2007.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1370

[3]. Bishop, C. M.*Pattern Recognition and Machine Learning*, Springer, 2006.
[4]. Goodfellow, I., Bengio, Y., & Courville, A.
[5]. *Deep Learning*, MIT Press, 2016. Pal, U., & Chaudhuri, B. B.“Indian Script Character Recognition: A Survey,” *Pattern Recognition*, Elsevier, 2004.
[6]. Arora, S., Bhattacharjee, D., Nasipuri, M., Basu, D. K., & Kundu, M. “Performance Comparison of SVM and ANN for Handwritten Devanagari Character Recognition,” *IJCSI*, 2010.
[7]. Jawahar, C. V., Kumar, M. N., & Ravindran, B. “A Bilingual OCR System for Hindi and Tamil,” *ICDAR Conference Proceedings*, 2007.
[8]. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. “Gradient-Based Learning Applied to Document Recognition,” *Proceedings of the IEEE*, 1998.