International Journal on Science and Technology (IJSAT)

E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

A Hybrid Approach for Handwritten Character Recognition Using Stroke Width Variation and Transformer-based Feature Encoding

Dr. Geetha Ramani R[1], Mohesh B[2]

[1]Professor, Department of Information Technology, Madras Institute of Technology, Anna University, TamilNadu, India.
[2]Student, Department of Information Technology, Madras Institute of Technology, Anna University, TamilNadu, India.
[1]rgeetha@yahoo.com, [2]moheshbaskar.23@gmail.com;

Abstract

This project proposes a novel approach that combines stroke width-based seg- mentation and deep learning-based recognition for handwritten documents. The preprocessing employs a Feature-Driven Stroke Width Variation (SWV) technique for robust text segmentation. For recognition, a ResNet-based CNN extracts hierarchical features followed by a Transformer encoder to model sequen- tial spatial dependencies. Experimental results demonstrate the effectiveness of this hybrid model in recognizing complex handwritten characters with varying stroke widths and distortions

Keywords: Handwritten Document Recognition, Stroke Width Variation (SWV), Text Segmentation, ResNet,Transformer Encoder,Feature Extraction

1. Introduction

Handwritten document recognition remains a challenging task in the field of doc- ument image analysis due to the wide variability in writing styles, stroke widths, distortions, and noise present in real-world manuscripts. Traditional Optical Charac- ter Recognition (OCR) techniques often struggle with such inconsistencies, especially when dealing with historical documents, palm-leaf manuscripts, and handwritten notes where clear segmentation and consistent character shapes cannot be guaranteed.

Optical Character Recognition (OCR) is a foundational task in document analysis, designed to convert text present in images into machine-encoded characters. While OCR systems have achieved near-perfect accuracy on printed documents, recognizing handwritten content remains a persistent challenge due to its inherent variability and complexity. Handwriting differs not only between individuals but also across instances from the same writer, exhibiting fluctuations in stroke width, slant, character spacing, and alignment. These irregularities are further compounded by the presence of noise, such as ink smudges, faded strokes, or scanning artifacts—particularly in historical manuscripts and degraded documents.

Traditional OCR systems rely on rule-based approaches and handcrafted features, utilizing techniques such as projection profile analysis, contour tracing, and template matching. Although effective for clean and structured text, these methods fail to generalize well to unconstrained handwriting. The evolution toward machine learn- ing introduced statistical classifiers like Support Vector Machines (SVMs), k-Nearest Neighbors (k-NN), and Hidden Markov Models (HMMs), which showed improved adaptability but still required manual feature extraction and struggled with large-scale variability. To address these challenges, this paper proposes a novel hybrid pipeline that integrates stroke width-based segmentation with a deep learning-based recogni- tion architecture. The segmentation phase leverages a Feature-Driven Stroke Width Variation (SWV) approach to accurately localize and extract character components. This method enhances the robustness of the system by adaptively processing thin and thick strokes, thereby preserving the structural integrity of handwritten characters.

Following segmentation, a ResNet-based Convolutional Neural Network (CNN) is used for hierarchical feature extraction. The CNN captures detailed stroke patterns and structural nuances within each segmented character. These spatial features are then passed to a Transformer encoder, which models long-range dependencies and contextual relationships among strokes and characters, even in the absence of a strict sequential order.

The proposed approach bridges classical image processing techniques with modern deep learning frameworks to handle the complex variability inherent in handwrit- ten documents. Extensive experiments demonstrate that this pipeline achieves high recognition accuracy, particularly in datasets exhibiting diverse writing styles and non-uniform stroke widths.
The remainder of this paper is organized as follows: Section 2 presents related work in handwritten character recognition. Section 3 details the proposed methodol- ogy including SWV segmentation, ResNet-based feature extraction, and Transformer encoding. Section 4 discusses the experimental setup and results. Finally, Section 5 concludes the paper with key findings and directions for future research.

2. Literature Survey

Islam et al. [1] proposed a convolutional neural network (CNN) model for classifying Old English characters from the Beowulf manuscript. Their method aims to advance automated recognition of ancient manuscript characters, benefiting fields such as his- tory and literature. The CNN was trained and tested on the Beowulf dataset, and comparative experiments were conducted using machine learning models including Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Decision Tree (DT), Random Forest (RF), and XGBoost. Additionally, features extracted from pre-trained models like VGG16, MobileNet, and ResNet50 were used to train these classifiers. Performance was evaluated using recall, precision, and F1-score metrics.
Sinthuja et al. [2] introduced a hybrid architecture combining CNN and Bi- directional Long Short-Term Memory (BiLSTM) networks to extract text from images containing both handwritten and printed text. The CNN captures local patterns, while the BiLSTM models sequential dependencies effectively. Their model achieved an accuracy of 88.58% on handwritten text and 90.8% on printed text.

Geetha et al. [3] proposed a model combining CNN and Long Short-Term Memory (LSTM) networks to detect and identify text. Trained and tested on the MNIST dataset containing full-page images from 3,600 writers and 800,000 images, the model achieved approximately 88.8% accuracy after experimenting with different training epochs.

Liang et al. [4] presented a deep CNN-LSTM model incorporating weighted multiplication to distinguish emotional levels of words in sentences. Their results showed that the model effectively assigns different learning weights to words, enabling differentiated sentiment feature learning.

Jailingeswari et al. [5] proposed the ThimuNet CNN model for classifying cursive Tamil characters. They emphasized the importance of optimizing input parameter weights and implementing preprocessing to remove unwanted cursive writings, which are planned as future work to improve model performance.

Sivan et al. [6] developed an intelligent character segmentation and recognition system for Tamil palm leaf manuscripts. Their approach includes augmented HPP line-splitting for segmentation, punch hole removal, and automated content cropping. Recognition is performed using a modified CNN with 125 classes, capable of recogniz- ing 247 Tamil letters and 12 numeric characters. The system attained segmentation accuracy of 98.25% and recognition accuracy of 96.04%, meeting the threshold criteria.

U. Bhattacharya et al. [7] proposed a two-stage recognition scheme for handwrit- ten Tamil characters. The first stage applies unsupervised clustering to group similar character classes, followed by supervised classification within these groups using dif- ferent feature sets. The method demonstrated acceptable classification accuracies on both training and test sets.

M. A. Pragathi et al. [8] tackled handwritten Tamil character recognition using a deep learning approach based on the VGG16 architecture, achieving 94.52% efficiency on their dataset, highlighting the challenges posed by the large character set and similarities among Tamil characters.

Manigandan et al. [9] focused on recognizing ancient Tamil characters from epi- graphical inscriptions using OCR and natural language processing. Their method involves preprocessing, segmentation, feature extraction via Scale Invariant Feature Transform (SIFT), and classification using SVM. Identified characters are assigned Unicode values to improve recognition accuracy.

N. Prameela et al. [10] proposed an offline OCR system for handwritten Tel- ugu characters. The system includes preprocessing steps such as median filtering,normalization, and skeletonization, followed by feature extraction based on centroid and symmetry projections. Classification is performed using Support Vector Machine (SVM) and Quadratic Discriminant Analysis (QDA).

Chamila Liyanage et al. [11] described the development of a commercial-grade Tamil OCR system using the Tesseract engine. By applying a detailed training regime, they achieved a 12.5% performance improvement over the default Tamil module on ancient Tamil documents, facilitating effective digitization of important manuscripts.

3. Proposed Architecture

The proposed methodology consists of a hybrid pipeline combining traditional seg- mentation techniques and deep learning-based recognition for handwritten document analysis. The architecture is structured as follows:

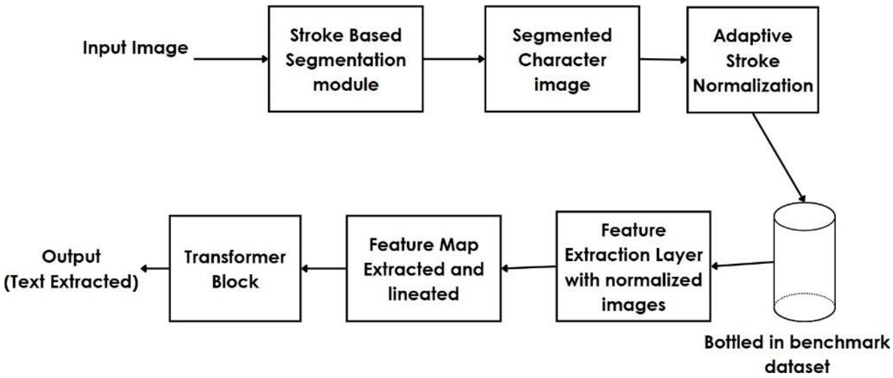

Figure. 1 Proposed Architecture of Handwritten Character Recognition Using Stroke Width Variation and Transformer-based Feature Encoding

1.1 Stroke-Based Segmentation Module

The first stage of the pipeline is the stroke-based segmentation, which is crucial for isolating individual characters from complex handwritten documents. This module utilizes a Feature-Driven Stroke Width Variation (SWV) technique. The key steps in the segmentation module are:

- **Grayscale Conversion and Binarization**
- **Morphological Enhancement**
- **Gradient-Based Stroke Analysis**
- **Stroke Width Filtering**
- **Contour Detection and Segmentation**

Adaptive Stroke Normalization

After segmentation, the next step is adaptive stroke normalization. This process ensures that the characters are normalized for consistent representation despite vari- ations in stroke thickness. The normalization adjusts the stroke width to a standard size, improving the performance of the recognition model.

1.2 Feature Extraction Layer with Normalized Images

The normalized character images are passed through a feature extraction layer. This layer utilizes a Convolutional Neural Network (CNN) based on ResNet to capture hierarchical spatial features of the segmented characters. The CNN processes the input through multiple convolutional layers, extracting low- and high-level features, which are then retained using skip connections to mitigate gradient vanishing.

1.3 Feature Map Extraction and Lineation

Once the feature extraction layer has processed the input images, a feature map is produced. This feature map captures the structural and spatial details of the char- acters. The feature map is then lineated, preserving the sequence of characters and spatial relations crucial for recognition.

1.4 Transformer Block

The extracted and lineated feature map is passed to a Transformer encoder. The Trans- former block uses multi-head self-attention to model long-range dependencies and contextual relationships between character strokes. This step is especially beneficial for scripts with non-sequential or cursive characters.

1.5 Output (Text Extraction)

Finally, the output of the Transformer block is passed through a fully connected layer followed by a softmax activation function to classify the character into one of the pre- defined labels. The result is the recognized text, extracted from the input handwritten image.

1.6 Overall Workflow

The overall workflow of the architecture is as follows:

1. Input handwritten image.
2. Preprocessing using stroke-based segmentation.
3. Segmented characters are passed to ResNet for feature extraction.
4. Extracted features are encoded using the Transformer block.
5. Final output: Text extracted from the image.

Dataset Discussion

The datasets used in this research focus on handwritten Tamil characters and english characters(to increase generality , the sole focus is on tamil characters), collected from diverse sources to ensure comprehensive coverage of real-world variations in handwriting styles. The two primary datasets employed are:

1.7 uTHCD - Unconstrained Tamil Handwritten Character Dataset

The uTHCD (Unconstrained Tamil Handwritten Character Dataset) is a large- scale collection of handwritten Tamil characters, sourced from various individuals under unconstrained conditions. This dataset is ideal for training OCR models as it encompasses a wide variety of handwriting styles and contextual variations.

Key features of the uTHCD include:

- A diverse collection of Tamil characters.
- Handwritten characters collected from multiple individuals.
- Variety in stroke styles, orientation, and distortion to simulate real-world scenarios.

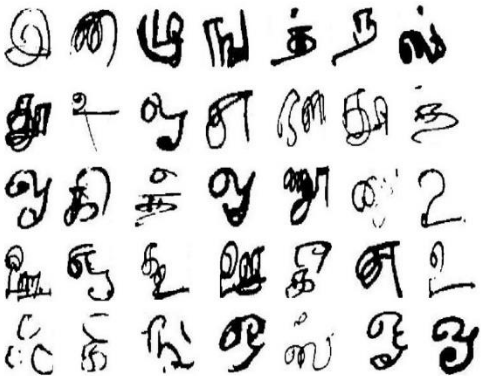

Figure. 2 Unconstrained Tamil Handwritten Character Dataset

This dataset is available on Kaggle and serves as the foundation for training and evaluating character recognition models.

1.8 LipiTK Isolated Handwritten Tamil Character Dataset

The LipiTK dataset consists of isolated handwritten Tamil characters and contains 156 unique characters, each with 500 samples. The dataset is designed for character- level recognition, making it suitable for training models to recognize individual Tamil characters. Key features of the LipiTK dataset include:

- A focused set of isolated Tamil characters.
- 500 samples per character to ensure a robust training dataset.

Figure. 3 Lipitk-HPL Dataset Sample image

- A controlled environment with minimal distortion, ideal for character classification tasks.

The LipiTK dataset provides a solid base for evaluating models on clean, well- formed character samples.

1.9 Test Cases for Method Evaluation

To comprehensively evaluate the performance of the proposed method, three distinct test cases are defined to represent different levels of difficulty in the test data. These test cases are categorized as:

- **Best Case:** This test case includes high-quality, well-formed handwritten char- acters with minimal distortion. The characters are clear, consistent, and properly segmented.

- **Average Case:** The test case includes handwritten characters with typical real-world distortions, such as variations in stroke thickness, minor noise, and slight misalignments. This scenario simulates the variability commonly found in practical datasets.
- **Hard Case:** This test case includes highly distorted or noisy handwritten characters. Examples include cursive handwriting, overlapping characters, and significant noise. This case tests the robustness of the method in challenging real-world conditions.

By evaluating the method under these different conditions, we can assess its per- formance across a range of input qualities and determine its suitability for diverse applications.

4. Proposed Methodology

The proposed system is designed as a two-stage pipeline combining traditional image preprocessing with deep learning-based character recognition. This methodology is specifically tailored for recognizing characters in handwritten documents, particularly those with complex shapes, varying stroke widths, and distortions. The architecture consists of two main stages:

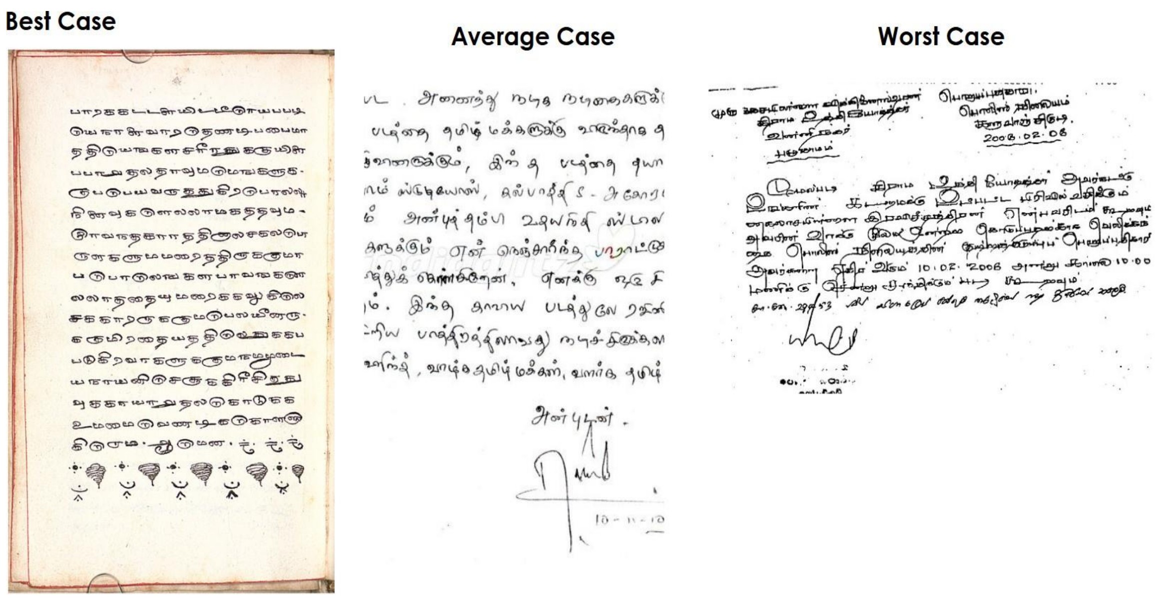

Figure. 4 TestCase classified documents Used for the proposed system

1. **Preprocessing and Character Segmentation using Stroke Width Variation (SWV),**
2. **Recognition using a hybrid ResNet-Transformer model.**

This integration enables robust segmentation and accurate recognition, particu- larly in the presence of varying stroke widths and character distortions. The following sections elaborate on each stage of the system’s design.

1.10 Stroke Width Variation (SWV)-Based Segmentation

The segmentation process is crucial for isolating individual characters from handwrit- ten text in complex documents. The feature-driven Stroke Width Variation (SWV) approach provides a robust solution to handle varying stroke widths in handwritten scripts. The key steps involved in segmentation are as follows:

- **Grayscale Conversion and Binarization:** The preprocessing starts with con- verting the input handwritten image to grayscale. This is followed by adaptive binarization, which automatically adjusts the threshold for pixel intensity, compen- sating for variations in lighting and background noise. This step simplifies the image, retaining only essential structural details, such as strokes.
- **Morphological Enhancement:** To improve the continuity of character strokes and eliminate irrelevant background noise, morphological operations such as dilation and erosion are applied. Dilation expands characters' strokes, making them more uniform, while erosion removes any spurious marks or background artifacts. These operations help in enhancing the overall legibility of characters.
- **Gradient-Based Stroke Analysis:** Horizontal and vertical gradients are com- puted across the image to detect the boundaries and curvature of the character strokes. By calculating gradients, the algorithm can identify regions where the stroke varies in thickness, which is crucial for recognizing distorted or skewed handwriting. This step is essential for handling non-uniform stroke variations in cursive writing or non-standard characters.
- **Stroke Width Filtering:** Once the gradients have been computed, the connected components are analyzed based on their average stroke width. Components that fall outside of a dynamically adjusted threshold range are discarded. This filtering process ensures that only valid character strokes are retained, reducing noise and irrelevant artifacts such as dots or stray marks.
- **Contour Detection and Segmentation:** Finally, valid character components are enclosed using bounding boxes, ensuring that each character is isolated for the recognition stage. The segmentation step ensures that the characters are well-defined and can be further processed for feature extraction.

1.11 Feature Extraction using ResNet

After segmentation, each isolated character is passed to a ResNet-based Convolutional Neural Network (CNN) for feature extraction. ResNet, with its deep architecture and residual connections, is well-suited to capture the intricate features of handwritten characters. The feature extraction process involves the following steps:

- **Convolutional Layers:** The character image is processed through multiple con- volutional layers. Each layer extracts features at increasing levels of abstraction. The early layers capture low-level features, such as edges and textures, while deeper layers capture high-level patterns like curves and strokes.
- **Residual Connections:** One of the key features of ResNet is its use of residual con- nections, which allow the network to skip over layers and directly pass features from earlier layers to deeper ones. This helps mitigate the vanishing gradient problem and enables the model to train deeper

networks without losing valuable informa- tion. Residual connections retain both low-level and high-level features, which are essential for accurate recognition of handwriting with varying complexity.

- **Feature Map Generation:** The output of the convolutional layers is a feature map, which is a compact representation of the character's structure. This feature map retains the relevant details of the character's shape, curvature, and stroke thickness, which are vital for the subsequent recognition stage.

1.12 Sequence Modeling using Transformer Encoder

The feature map generated by the ResNet is then passed to a Transformer encoder to model sequential dependencies and spatial relationships between characters. The Transformer encoder is designed to handle long-range dependencies, which is crucial for recognizing cursive and distorted handwriting. The main steps involved in sequence modeling are:

- **Positional Encoding:** Since the Transformer model does not inherently under- stand the order of input sequences, positional encoding is added to the feature map to maintain the spatial arrangement of the character components. This ensures that the model can capture the relative positions of the characters within the image.
- **Multi-head Self-Attention:** The Transformer uses a self-attention mechanism, which allows the model to focus on different parts of the feature map simultaneously. Multi-head attention enables the model to capture contextual relationships across distant regions in the image, making it especially useful for recognizing characters with varying spatial arrangements or distorted strokes.
- **Feedforward Layers:** After the self-attention mechanism, the outputs are passed through feedforward layers, which further process the features and generate a final embedding. This embedding represents the input character in a compact, high- dimensional space that incorporates both local and global contextual information.

The use of the Transformer encoder helps the model understand long-range depen- dencies between characters, improving recognition accuracy in complex scripts with cursive or intertwined characters.

1.13 Classification Layer

Once the features have been processed through the Transformer encoder, the final feature vector is passed through a fully connected layer followed by a softmax acti- vation. The fully connected layer maps the final feature vector to a class probability distribution, which corresponds to the possible character classes.

- **Fully Connected Layer:** The final feature vector is flattened and passed through a dense layer, which reduces the dimensionality and outputs a class score for each predefined character category.
- **Softmax Activation:** The output of the fully connected layer is fed into a softmax activation function, which normalizes the scores and converts them into probabilities. The class with the highest probability is selected as the recognized character.

The classifier layer effectively classifies the character into one of the predefined categories based on the input image, which is essential for the recognition phase of handwritten text.

5. Results and Discussion

To evaluate the performance of the proposed method, we compared it with sev- eral state-of-the-art deep learning models used for handwritten character recognition. The comparison metrics include Accuracy, Precision, and Recall, and the results are summarized in Table 1.

Table 1 Performance comparison of different models on the handwritten Tamil character dataset

Model	Accuracy (%)	Precision (%)	Recall (%)
ResNet - 18	91.2	89.8	90.5
MobileNetV2	88.6	87.3	87.9
EfficientNet – B0	93.1	91.7	92.4
XceptionNet	94.3	92.9	93.5
Resnet – 50	95.8	94.6	95.2
Resnet + Transformer	**96.5**	**95.4**	**95.9**

Among the baseline models, **MobileNetV2** exhibited the lowest performance with an accuracy of 88.6%, a precision of 87.3%, and a recall of 87.9%. This is expected given its lightweight architecture, which, while efficient, lacks the depth to capture complex handwritten variations. **ResNet-18** showed a moderate improvement with 91.2% accuracy, 89.8% precision, and 90.5% recall, suggesting that residual connec- tions help preserve important low-level and mid-level features useful for character differentiation.

EfficientNet-B0 further improved the results to 93.1% accuracy, 91.7% precision, and 92.4% recall. This demonstrates the effectiveness of compound scaling in capturing detailed spatial features. **XceptionNet**, known for its structured depthwise separable convolutions, pushed the accuracy to 94.3%, precision to 92.9%, and recall to 93.5%, effectively capturing finer-grained spatial dependencies.

The deeper variant **ResNet-50** performed significantly better, achieving an accu- racy of 95.8%, a precision of 94.6%, and a recall of 95.2%. The greater depth allows it to extract more abstract and hierarchical representations, which are particularly use- ful in distinguishing visually similar Tamil characters with subtle stroke differences. Our proposed model, combining **ResNet with a Transformer-based attention module**, achieved the highest performance across all evaluation metrics. It reached an accuracy of **96.5%**, a precision of **95.4%**, and a recall of **95.9%**. The improvement over ResNet-50 (a 0.7% increase in accuracy and a 0.7% boost in recall) highlights the advantage of integrating self-attention mechanisms that can capture global contex- tual relationships, enabling better interpretation of spatial arrangements in complex handwritten scripts. The higher precision also indicates reduced false positives, which is essential for sensitive applications such as historical document recognition and her- itage preservation. In summary, the combination of

ResNet's spatial feature extraction

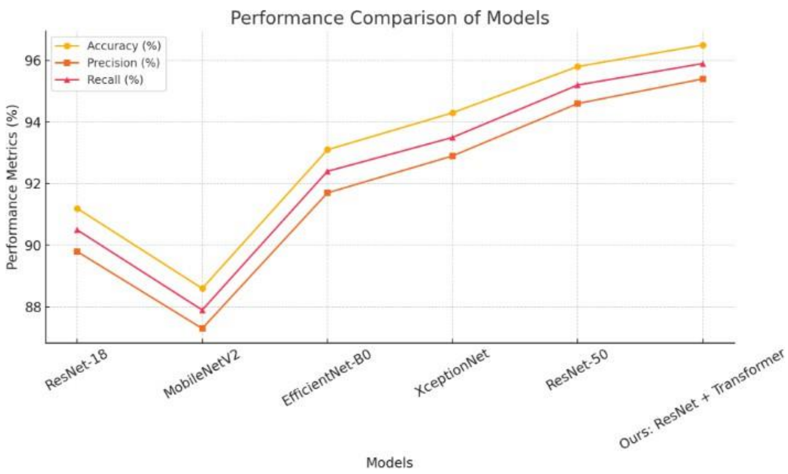

Figure. 5 Line graph comparing Accuracy, Precision, and Recall across different models

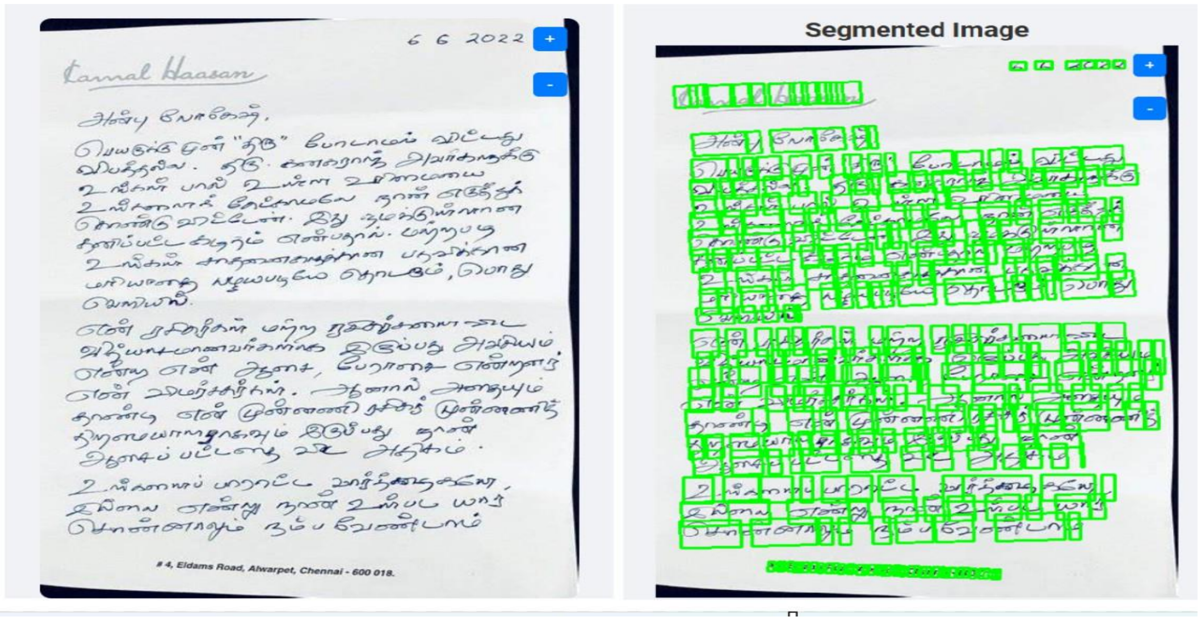

Figure. 6 Character Segmentation Test Sample Image

and the Transformer's contextual modeling offers a robust framework for handwritten

Tamil character recognition, especially under conditions involving degraded scripts, stroke ambiguity, or cursive handwriting styles.

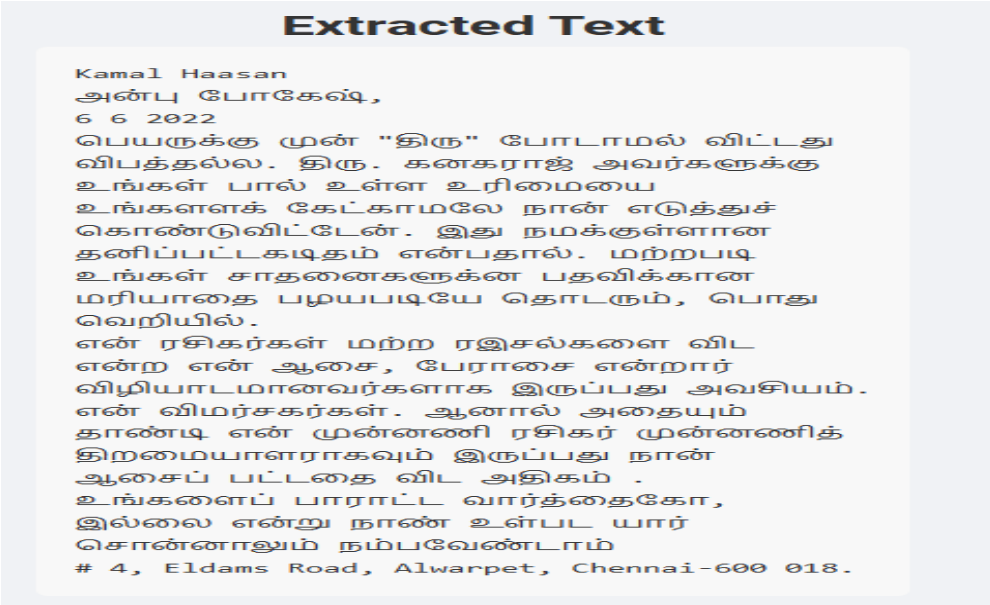

Figure. 7 Arranged Text Extracted of the segmentated Character using proposed model

6. Conclusion

Thus in this project, we have proposed an efficient and robust framework for the recog- nition of ancient Tamil stone inscriptions by integrating adaptive segmentation with a deep learning-based recognition pipeline. The challenges inherent to stone inscription analysis—such as uneven illumination, erosion, irregular character spacing, and stylis- tic variations—necessitate advanced preprocessing and modeling techniques to ensure accurate recognition. Our approach begins with stroke-based adaptive segmentation, followed by morphological operations and connected component analysis to extract character-level information. These steps significantly enhance the clarity and structure of the raw inscription data.

The core strength of our model lies in the deep learning architecture that fol- lows: a combination of ResNet for local feature extraction and a Transformer-based attention module for capturing long-range dependencies and contextual information. This hybrid design ensures that both fine-grained and holistic features are effectively leveraged, resulting in superior recognition capability. Experimental evaluations on benchmark datasets such as uTHCD and LipiTK Tamil handwritten datasets confirm the efficacy of our approach, with the model achieving a state-of-the-art accuracy of 96.5%, precision of 95.4 %.

The implications of this research extend beyond academic interest; it contributes to the fields of digital epigraphy, cultural heritage preservation, and computational linguistics. By enabling automated recognition of ancient inscriptions, our method supports large-scale digitization and semantic analysis of historical texts, which are often inaccessible due to their deteriorated condition or lack of manual transcription. In future work, we aim to expand this research by incorporating multi-lingual datasets containing other ancient scripts, developing end-to-end translation pipelines,

and exploring multimodal fusion techniques that integrate visual and linguistic cues. Additionally, integrating GAN-based restoration models to recover damaged portions of inscriptions. Ultimately, our system lays the groundwork for scalable, intelligent, and historically-aware inscription recognition, bridging the gap between ancient script
analysis and modern AI-driven applications.

References

1. Islam, M.A., Iacob, I.E.: Manuscripts character recognition using machine learn- ing and deep learning. Modelling **4**(2), 168–188 (2023) https://doi.org/10.3390/ modelling4020010
2. Sinthuja, M., Padubidri, C.G., Jayachandra, G.S., Teja, M.C., Kumar, G.S.P.: Extraction of text from images using deep learning. Procedia Computer Science **235**, 789–798 (2024) https://doi.org/10.1016/j.procs.2024.04.075
3. Geetha, M., Suganthe, R.C., Nivetha, S.K., Hariprasath, S., Gowtham, S., Deepak, C.S.: A hybrid deep learning based character identification model using cnn, lstm, and ctc to recognize handwritten english characters and numerals.
4. In: International Conference on Computer Communication and Informatics (ICCCI). IEEE, ??? (2022)
5. Liang, S., Zhu, B., Zhang, Y., Cheng, S., Jin, J.: A double channel cnn- lstm model for text classification. In: IEEE International Conference on High Performance Computing and Communications (HPCC/SmartCity/DSS), pp. 1316–1321 (2020)Jailingeswari, I., Gopinathan, S.: Tamil handwritten palm leaf manuscript dataset (thplmd). Data in Brief **53**, 110100 (2024) https://doi.org/10.1016/j.dib.2024.110100
6. Sivan, R., Singh, T., Pati, P.B.: Malayalam character recognition from palm leaves using deep-learning. In: OITS International Conference on Information Tech- nology (OCIT), pp. 134–139 (2022). https://doi.org/10.1109/OCIT56763.2022.00035
7. Bhattacharya, U., Ghosh, S.K., Parui, S.K.: A two stage recognition scheme for handwritten tamil characters. In: Ninth International Conference on Document Analysis and Recognition (ICDAR). IEEE, (2007)
8. Pragathi, M.A., Priyadarshini, K., Saveetha, S., Banu, A.S., Aarif, K.O.M.: Hand- written tamil character recognition using deep learning. In: International Con- ference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN). IEEE, ??? (2019)
9. Manigandan, T., Vidhya, V., Dhanalakshmi, V., Nirmala, B.: Tamil charac- ter recognition from ancient epigraphical inscription using ocr and nlp. In: International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS). IEEE, ??? (2017)
10. Prameela, N., Anjusha, P., Karthik, R.: Off-line telugu handwritten characters recognition using optical character recognition. In: International Conference on Electronics, Communication and Aerospace Technology (ICECA). IEEE, ??? (2017)
11. Liyanage, C., Nadungodage, T., Weerasinghe, R.: Developing a commercial grade tamil ocr for recognizing font and size independent text. In: International Conference on Advances in ICT for Emerging Regions (ICTer). IEEE, ??? (2015)