International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(5s) 2026 pp. 944-960
https://doi.org/10.70917/ijcisim-2026-1860
Article

A Hybrid CNN–Capsule–Transformer Architecture for Indic Handwritten Text Recognition with Cross-Script Evaluation

A S Manjunath[1], Umesh Dadadahalli Ramu[2], Madhusudan G[3], Meghana J[1,*]

[1]Department of Computer Applications, JSS Science and Technology University, India
[1] Visvesvaraya Technological University, Belagavi
[2]Department of Computer Science and Engineering, PES College of Engineering, Mandya, India
[3]Department of Computer Science and Engineering, JSS Science and Technology University, India
[1]Department of Computer Applications, JSS Science and Technology University, India
[1]as.manjunath@jssstuniv.in,[2]umesh.dr.pesce@gmail.com,[3]madhusudan@sjce.ac.in,[4]meghanaj@jssstuniv.in
***Corresponding author:** Meghana J, meghanaj@jssstuniv.in

Abstract: Recognition of Indic handwritten text is a difficult issue with the complex formations of characters, variability of graphemes, ambiguity of strokes, and significant differences between writers. This is particularly problematic in scripts (such as Tamil and Kannada) where the form of the handwritten words and the composition structure often are not regular. Although recent CNN-RNN and CNN-Transformer architecture have achieved encouraging results, they either pay much attention to local visual representation or global context representation and do not consider the structural relationship of handwritten patterns at an appropriate level. This work will offer a solution to this drawback by suggesting a Hybrid CNN 10 Capsule 10 Transformer network with Connectionist Temporal Classification (CTC) to perform end-to-end handwritten word recognition. The proposed framework uses CNN layers to extract local visual features, the capsule module to encode structural and compositional relationships, and the Transformer to learn long-range sequence dependencies to be correctly transcribed. The model is tested on Tamil and Kannada handwritten data to check the effectiveness of cross-scripts. The results of the experiment indicate that the given architecture has a test accuracy of 85.46 percent and a Character Error Rate (CER) of 0.0281 on Tamil and 88.70 percent and a Character Error Rate (CER) of 0.0175 on Kannada. These results indicate that the hybrid framework proposed enhances the performance of cross-script handwritten text recognition as compared to baseline architectures.

Keywords: Handwritten text recognition, Indic script recognition, CNN–Capsule–Transformer, Connectionist Temporal Classification, cross-script evaluation, Tamil, Kannada.

1. Introduction

Due to its usefulness in archival digitization, managing educational records, automating administrative functions, and intelligent understanding of documents, handwritten text recognition (HTR) has emerged as a significant research field in document analysis. In contrast to printed text recognition, HTR needs to address irregular styles of writing, deformation of strokes, dissimilarity in character sizes, intertwined elements, noise, and unreliable spacing. The key advancement in sequence-based recognition was the Connectionist Temporal Classification which facilitated the prediction of labels on unsegmented sequential data without explicit character-level alignment [1]. This was succeeded by multidimensional recurrent neural network methods of offline handwriting recognition that showed that spatial context and sequence modeling can be synergistically utilized in handwritten word transcription [2]. Subsequently, the use of end-to-end trainable convolutional-recurrent architectures enhanced image-based sequence recognition by integrating the extraction of visual features with sequential decoding, which formed a solid baseline to the current text recognition pipelines [3].

The developments are especially important to Indic handwritten text recognition, the scripts of which are both visually and structurally complicated. Compound constructions, modifiers, curved lines, and considerable intra-class deviation due to personal styles of writing are common features of Indic scripts. Gated convolutional recurrent architectures in multilingual handwriting recognition The model demonstrates that script diversity poses an additional modeling challenge on top of the traditional text recognition problems [4]. Meanwhile, research into the need of recurrent layers of an even greater complexity has proposed that other strategies in architecture can provide a competitive or even superior performance, especially joint learning of a better visual and sequential representation [5]. Here, the strong feature extraction is also critical, and residual learning models have significantly contributed to the empowerment of deep visual representations to the recognition-based vision tasks [6].

Although this has been achieved, the current HTR methods have weaknesses with complex handwritten Indic words. CNNRNN models are useful to capture local visual representations and sequence dynamics, but they might fail when it comes to long-distance and delicate structural interactions in handwritten words. Transformer-based models mitigate this weakness in part, by modelling global contextual dependencies with self-attention to make a more effective use of purely recurrent models [7]. Nevertheless, CNN-Transformer pipelines run in their pure form can still fail to represent part-whole hierarchies that are salient in handwritten scripts. In this respect, capsule networks are an effective alternative option since they encode features as vectors and ensure spatial constraints between learned elements, which is why they are applicable to structurally complex visual recognition problems [8].

Inspired by these shortcomings, this paper creates a hybrid recognition architecture that combines the advantages of convolutional, capsule-based, and Transformer-based learning. The coding structure in this paper is based on this rationale, in CNN layers the local stroke and texture features are extracted using the images of handwritten words, the capsule module is in charge of preserving the association between structural and compositional features, and the Transformer encoder is in charge of capturing the long-range dependencies across the width-wise representation of the word image. It is specifically applicable to the case of Indic handwritten text recognition, wherein the correct prediction cannot be achieved without structurally sensitive as well as context-sensitive modeling along with local pattern extraction. Moreover, the research also compares the architecture of two Indic scripts, Tamil and Kannada, to determine whether the same design can be effective with scripts that have visual and graphemic properties different. This study has the following main objectives and contributions:

- To develop a Hybrid CNN–Capsule–Transformer architecture for Indic handwritten word recognition by integrating convolutional feature extraction, capsule-based structural encoding, and Transformer-based sequence modeling.
- To evaluate the proposed framework on Tamil and Kannada handwritten datasets and perform cross-script assessment using the same architectural design.
- To compare the proposed model against CNN + BiLSTM + CTC and CNN + Transformer + CTC baselines, and to validate its effectiveness through word accuracy, Character Error Rate (CER), and error analysis.

2. Literature Review

Transformer-based handwritten text recognition has been of great interest due to its capability to represent long-range dependencies in a better way than traditional recurrent methods. One of the significant studies in this direction was done by [9] who proposed the framework of Transformer to recognize handwritten text with the help of the post-decoding (bidirectional) approach. Their experiments demonstrated a contextual refinement process in both forward and backward decoding directions that may enhance the quality of transcription particularly on the sequence-sensitive handwritten text tasks. This research has some relevance since it determined that sequence modeling based on attention could have a powerful role to play in handwritten recognition as opposed to standard recurrent pipelines.

Barrere et al. [10] took this path further and suggested a light Transformer-based text recognition system that recognized handwritten text. Their work highlighted the efficiency in computations but maintaining recognition accuracy, proving that the Transformer-based HTR systems do not always have to be too large or computationally intensive. This plays a significant role in real-world applications, especially when efficiency of the models and their ability to be deployed have to be taken into account in addition to predictive performance.

Li et al. [11] also presented TrOCR, a Transformer-based character recognition framework that is based on pre-trained models and operates on an optical character recognition task. The importance of this work is that it showed that, when provided with large-scale pretraining, recognition can be greatly enhanced through transferring learned representations between OCR-related tasks. Even though TrOCR is not limited to the offline handwritten text

recognition alone, it is highly encouraging the increasing role of Transformer-based models in the current text recognition studies.

Parres et al. [12] also made a step forward in this field by exploring the recognition of handwritten documents with pre-trained vision transformers. Their research point out that visual Transformer backbones could be very useful in the extraction of deep and rich features of handwritten documents. The book is of particular use, as it relates handwritten document recognition to the overall success of vision transformers in computer vision and pattern recognition.

Retsinas et al. [13] suggested to add Transformer blocks to CRNN-based handwritten text recognition architectures. Instead of fully substituting the old sequence-based models, their research demonstrated that these hybrid frameworks of convolutional-recurrent frameworks and attention-based modules can be used to enhance the quality of recognition. This is exceptionally applicable to the current research since it upholds the greater concept that hybrid designs would work better than standalone modeling plans, as they would jointly make up strengths.

Li et al. [14] presented the HTR-VT, which is a framework of vision transformer and recognizes handwritten text. Their study established that Transformer-based visual modeling is now a promising study line in HTR and can perform competitively without the need to solely use the traditional recurrent decoding. This research is significant because it will support the use of Transformer-based architectures as an up-to-date option in sequence recognition in handwriting applications.

Data set development and benchmark creation is also critical, in the point of view of Indic handwritten text recognition. Gongidi and Jawahar [15] presented the IIIT-Indic-HW-Words dataset that offered a specific benchmarking tool to recognize Indic handwritten words. This input is especially important since a strong experimentation of Indic HTR needs well-curated datasets that capture script and writing variation and realistic complexity of handwritten text.

Lalitha et al. [16] concentrated on the particular aspect of increasing accuracy in Indic handwritten text recognition. Their effort pointed out that Indic scripts are especially difficult due to the character composition, structure and the difference in the performance of the handwritten characters. The work is immediately applicable as it supports the idea of greater recognition requirements based on the needs of Indic handwriting instead of depending solely on approaches that have been tested on Latin-based data.

The general problem of resource constraints in offline handwritten text recognition was tackled in a study by Mondal et al. [17]. Their work highlighted that the quality of recognition can not only be affected by the design of the model but also by the availability of benchmarks, the diversity of data and training. Though their research concentrated on offline English handwritten recognition, the implication of the research is relevant to Indic handwritten recognition since the same restrictions in data and system design may impact the performance.

Tulsyan et al. [18] suggested a benchmark system to Indian language text recognition and has provided a valuable comparative baseline to language-specific and script-aware recognition systems. This contribution can be important in the sense that benchmark systems can be used to make a structured comparison between architectures and used as a reference point to assess the progress of Indian language text recognition research.

Dhiaf et al. [19] proposed a lightweight Transformer model called MSDocTr-Lite to recognize multi-script handwriting on full-page. Their analysis showed that Transformer-based systems can be generalized to more complex handwriting contexts of full-page and multi-script recognition. This contribution is significant as it indicates that lightweight attention-based architecture can be used to achieve scalable handwritten recognition with the script diversity. L. S. Anusha et al. [21] Experimental results indicate that heavily degraded images exhibit PSNR values below 20 dB and SSIM values below 0.1. Such images contain significant noise and poor visual quality, making text recognition more challenging. Therefore, they require more intensive preprocessing compared to higher-quality images with lower noise levels.

Collectively, these studies show that there is an evident development in the literature on handwritten text recognition. Recent studies have shifted to Transformer-based and hybrid pipelines and recurrent and CRNN-based pipelines, and parallel developments in dataset creation, and benchmarking have bolstered experimentation in Indic and multi-script contexts. Nonetheless, there exists a literature gap where architectures have been developed that combine local visual feature extraction, structurally aware encoding with global contextual sequence modeling to recognize Indic handwritten words. The gap informs the work at hand, which builds upon and analyses a Hybrid CNN-Capsule-Transformer architecture to Tamil and Kannada handwritten text recognition.

3. Methodology

3.1 Research Workflow Overview

The suggested framework uses a systematic pipeline in the recognition of handwritten text, which combines the extraction of features, hierarchical encoding, sequence modeling, and transcription. First, the image words are input and initially pre-processed and normalized to a constant resolution to provide consistency across samples. A convolutional neural network (CNN) backbone is then used to extract spatial features with these processed images. The resulting feature maps are again optimized with the help of a capsule-based representation module which includes the spatial structures and part-whole structures in handwritten characters.

The encoded features are subsequently translated into sequential representations and then a Transformer encoder is applied to extract long-range dependencies amongst character sequences. Lastly, a Connectionist Temporal Classification (CTC) layer is used to transform the sequence output to the desired text, without necessarily matching characters. The architecture diagram shows the general workflow which consists of the combination of CNN, capsule and Transformer modules into one recognition pipeline.

3.2 Datasets and Data Partitioning

The experimental evaluation was conducted on two Indic handwritten word datasets, namely Tamil and Kannada [20]. The Tamil dataset was partitioned into 75,736 training samples, 11,598 validation samples, and 16,184 test samples. The Kannada dataset was partitioned into 73,517 training samples, 13,752 validation samples, and 15,730 test samples. In both cases, the experiments were performed using predefined train, validation, and test splits in order to maintain consistency and ensure fair model comparison.

These datasets contain grayscale handwritten word images with substantial variation in writing style, stroke formation, and character composition. Such variability makes them suitable benchmarks for evaluating the robustness of handwritten text recognition systems. The same data partitioning protocol was maintained across all compared models, including the proposed hybrid model and the baseline architectures.

3.3 Data Preprocessing and Text Representation

The input images are scaled to a standard size and normalized to increase training stability. The images are padded to a constant width to fit the variable-length handwritten words, and aspect ratios are maintained. This makes it compatible with batch processing in the course of training.

The encoding of text labels is done in the form of character or grapheme-level representations. In the case of Tamil, it uses grapheme cluster extraction to efficiently encode intricate character structures, and character-level encoding to encode Kannada text. The vocabulary of sequence prediction is a mapping of each distinct symbol to an integer index. Sequences are padded and batched with a custom collate function during training, and effective input lengths needed to compute the CTC loss are also computed.

3.4 Baseline Models

In order to measure the effectiveness of proposed hybrid architecture, two baseline models are put into place. The initial baseline is comprised of CNN feature extractor and then a bidirectional Long short term memory (BiLSTM) network and CTC decoding layer. In this model sequential dependencies are captured with recurrent processing.

The second baseline substitutes the recurrent part with a Transformer encoder, and it creates CNN Transformer CTC. This model makes use of self-attention to model sequences. Both baselines are trained in the same conditions and thus can be fairly compared to the proposed hybrid model.

3.5 Proposed Hybrid CNN–Capsule–Transformer Architecture

The proposed model consists of three main components: a convolutional feature extractor, a capsule-based structural encoder, and a Transformer-based sequence modeling module, followed by a CTC prediction layer.

3.5.1 CNN Feature Extraction Module

The model begins with a convolutional backbone composed of three convolutional blocks. The first block applies a 3×3 convolution with 64 output channels, followed by batch normalization, ReLU activation, and 2×2 max pooling. The second block applies a 3×3 convolution with 128 output channels, followed by batch normalization,

ReLU activation, and 2 × 2 max pooling. The third block applies a 3 × 3 convolution with 256 output channels, followed by batch normalization and ReLU activation. This module extracts local stroke-level and texture-based features from handwritten word images.

3.5.2 Capsule Feature Encoding Module

The CNN feature maps are passed to a capsule-based encoding module to capture structural relationships between features. The capsule block applies a 3 × 3 convolution and reshapes the features into 8 capsules, each with a dimensionality of 16. A squash activation function is applied to normalize capsule vectors and preserve part–whole relationships. This enables better modeling of compositional structures in Indic scripts.

3.5.3 Transformer Sequence Modeling Module

The capsule output is converted into a sequential representation along the width dimension. After flattening, a linear projection maps the features into a 256-dimensional embedding space. Positional encoding is added to preserve sequence order. The sequence is then processed using a Transformer encoder with 4 layers, 8 attention heads, and a feed-forward dimension of 512. This module captures long-range dependencies in handwritten text.

3.5.4 CTC-Based Prediction Layer

The Transformer output is passed through a linear classification layer to generate token probabilities for each time step. The model is trained using Connectionist Temporal Classification (CTC) loss, which enables alignment-free sequence prediction. During inference, greedy decoding is applied by removing repeated tokens and blank symbols.

3.6 Training Configuration

The models are trained with CTC loss as the objective function by Adam optimizer. Training is done in multiple epochs with fixed batch size and the models are trained using backpropagation. It is applied in a deep learning model that is accelerated with a GPU in order to make it computationally efficient. The checkpoints are stored in training to verify the performance in between epochs. The CTC loss function was employed for sequence prediction. To ensure reproducibility, the random seed was fixed at 42 across Python, NumPy, and PyTorch. The implementation was executed on a CUDA-enabled GPU.

3.7 Evaluation Metrics

Character Error Rate (CER) and word-level accuracy are used to measure model performance. CER is a normalized edit distance statistic between the predicted and ground-truth sequences which can provide a fine-grained character-level error measure. Word accuracy is a measurement of correctly predicted words, and a general measure of performance. A set of these measures allows completely evaluating the accuracy of recognition of different scripts.

4. Results and Analysis

4.1 Performance Comparison Across Models

This subsection compares the overall effectiveness of the proposed hybrid architecture with the baseline models in the two Indic scripts. The task is to determine whether the combination of CNN, capsule, and Transformer elements can result in an improvement in recognition accuracy and error reduction.

The relative performance over the Tamil and Kannada datasets are summarised in Table 1 which documents the validation and test measures in Character Error Rate (CER) and word correctness. The findings are clear that the hybrid model is the lowest in terms of CER and highest in terms of accuracy in the two datasets.

In the case of Tamil, it is the hybrid model that brings the test CER to 0.0281 and the accuracy of 85.46, which is better than CNN-BiLSTM and CNN-Transformer baselines. Equally, in the case of Kannada, the model is able to attain a CER of 0.0175 and a 88.70 accuracy with a more notable increase over baselines. The quantitative analysis of all the models, as well as both datasets, is summarized in Table 1.

Table 1. Multilingual Performance Comparison of Baseline and Proposed Models on Tamil and Kannada Datasets

Language	Model	Val CER	Val Accuracy	Test CER	Test Accuracy
Tamil	CNN + BiLSTM + CTC	0.0338	0.8350	0.0337	0.8295
Tamil	CNN + Transformer + CTC	0.0354	0.8308	0.0344	0.8282
Tamil	Hybrid CNN + Capsule + Transformer + CTC	0.0286	0.8557	0.0281	0.8546
Kannada	CNN + BiLSTM + CTC	0.0248	0.8461	0.0242	0.8455
Kannada	CNN + Transformer + CTC	0.0275	0.8286	0.0271	0.8313
Kannada	Hybrid CNN + Capsule + Transformer + CTC	0.0171	0.8905	0.0175	0.8870

As Table 1 indicates, the proposed hybrid architecture outperforms the other set-ups in the two scripts. In the case of Tamil, the model enhances the accuracy of the test to 85.46% with a test CER of 0.0281. In the case of Kannada, the model also increases the accuracy of the test to 88.70% and the test CER to 0.0175. These improvements over both baseline architectures show that the hybrid architecture is useful in enhancing both exact-sequence recognition and character-level fidelity.

4.1.1 Comparative Visualization of Model Performance

Although Table 1 proves the numerical superiority of the proposed architecture, graphical comparisons are used to unveil the distribution and consistency of the improvements in a more obvious way. Specifically, visual plots facilitate easier comparison of validation and test trends across the architectures within any given script. In the case of the Tamil data, the comparison in terms of the model is depicted in Figure 1.

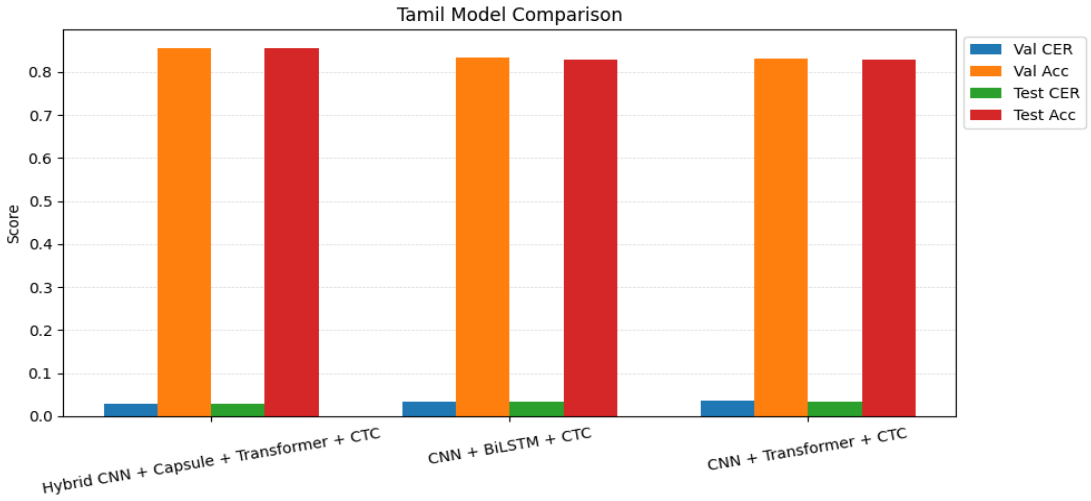

Figure 1. Tamil Model Comparison Across Architectures.

Test CER and accuracy comparison CNN + BiLSTM + CTC, CNN + Transformer + CTC and Hybrid CNN + Capsule + Transformer + CTC on the Tamil dataset.

As shown in Fig. 1, the hybrid model ensures the minimal error rates and maximum accuracy in both the test and validation conditions. The enhancement is not confined to a metric, which implies that the architecture improves

sequence recognition and accuracy in transcription. The same is also compared in Figure 2 corresponding to the Kannada dataset.

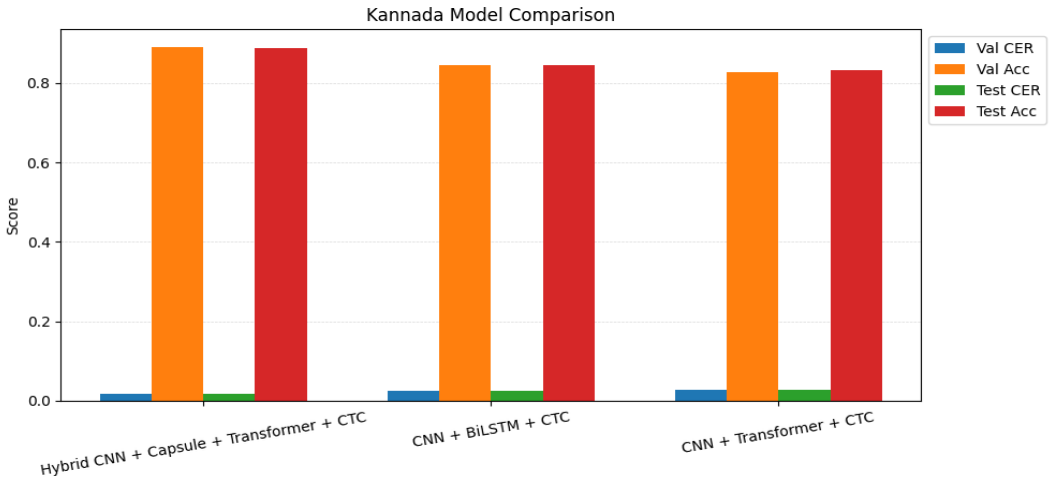

Figure 2. Kannada Model Comparison Across Architectures.

Checking and test CER and accuracy comparison with the three architectures tested on the Kannada dataset. Figure 2 shows a similar but even stronger trend of Kannada. The gaps between the hybrid model and the baselines are greater, especially in CER, which means that the proposed architecture is particularly efficient in minimizing character-level recognition errors in Kannada handwriting.

4.2 Relative Performance Improvement Analysis

Absolute performance values determine the best model, but do not entirely reveal the degree of improvement brought by the suggested design. This subsection thus estimates the relative benefits of the hybrid architecture compared to the baseline models. The improvements are measured in terms of percentage increase in test accuracy and percentage reduction in test CER.

4.2.1 Relative Improvement on Tamil Dataset

In the case of Tamil, the relative gains are calculated against the hybrid model directly against the CNN + BiLSTM + CTC and CNN + Transformer + CTC baselines. Through this analysis, one can be able to conclude on whether the hybrid model has only slight advantages or a significant enhancement over the prevailing architectures. Relative improvements of Tamil are shown in Figure 3.

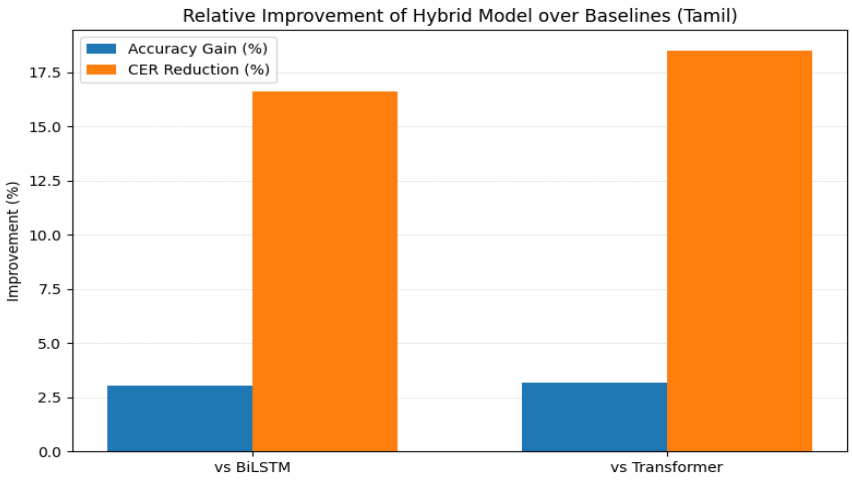

Figure 3. Relative Improvement of Hybrid Model over Baselines (Tamil).

Percentage accuracy gain and CER reduction of the hybrid model relative to the baseline architectures on the Tamil dataset. Figure 3 shows that the hybrid architecture provides measurable gains in both word accuracy and CER reduction. Although the increase in word accuracy is moderate, the reduction in CER is more pronounced, indicating that the architecture is particularly effective at minimizing fine-grained character-level errors in Tamil.

4.2.2 Relative Improvement on Kannada Dataset

The same relative improvement analysis was performed for Kannada to assess whether the benefits of the hybrid architecture are script-dependent or remain consistent across datasets. The improvement trends for Kannada are presented in **Figure 4**.

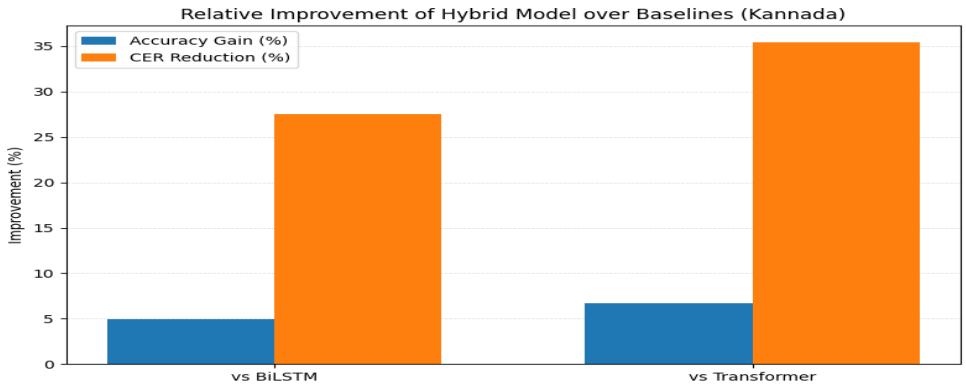

Figure 4. Relative Improvement of Hybrid Model over Baselines (Kannada).

Percentage accuracy gain and CER reduction of the hybrid model relative to the baseline architectures on the Kannada dataset. Figure 4 indicates that the performance gains are stronger in Kannada than in Tamil. In particular, the CER reduction is substantial, suggesting that the architecture is highly effective in capturing character structures and reducing transcription errors in Kannada handwriting.

4.2.3 Cross-Script Comparison of Relative Gains

To compare the magnitude of improvement across both scripts in a consolidated manner, the relative gains in accuracy and CER were also visualized at the cross-script level. The cross-script comparison of test accuracy gains is shown in **Figure 5**, while the corresponding CER reduction is shown in **Figure 6**.

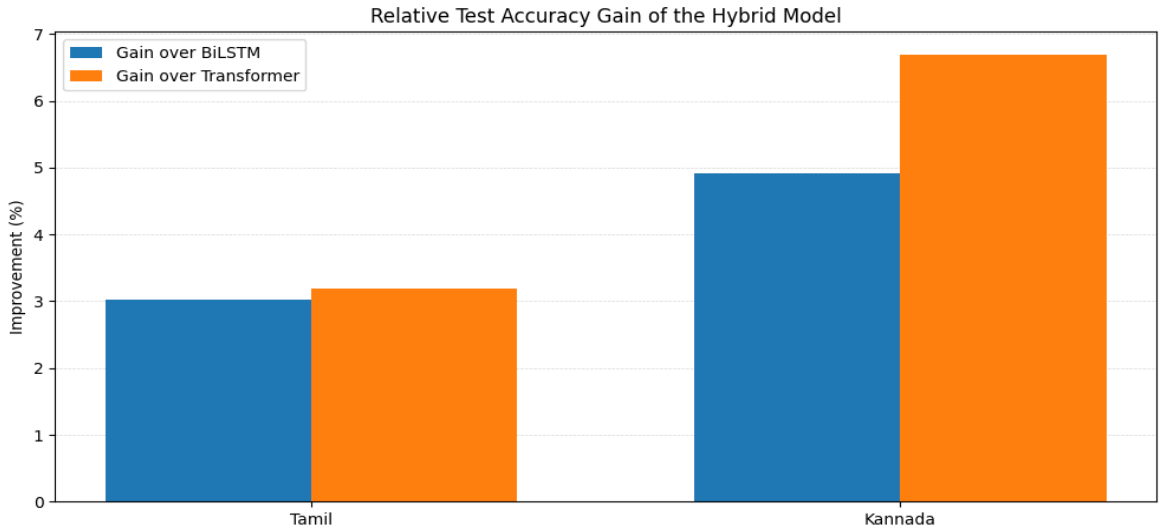

Figure 5. Relative Test Accuracy Gain of the Hybrid Model Across Scripts.

Comparison of test accuracy gain achieved by the hybrid model over baseline models for Tamil and Kannada datasets.

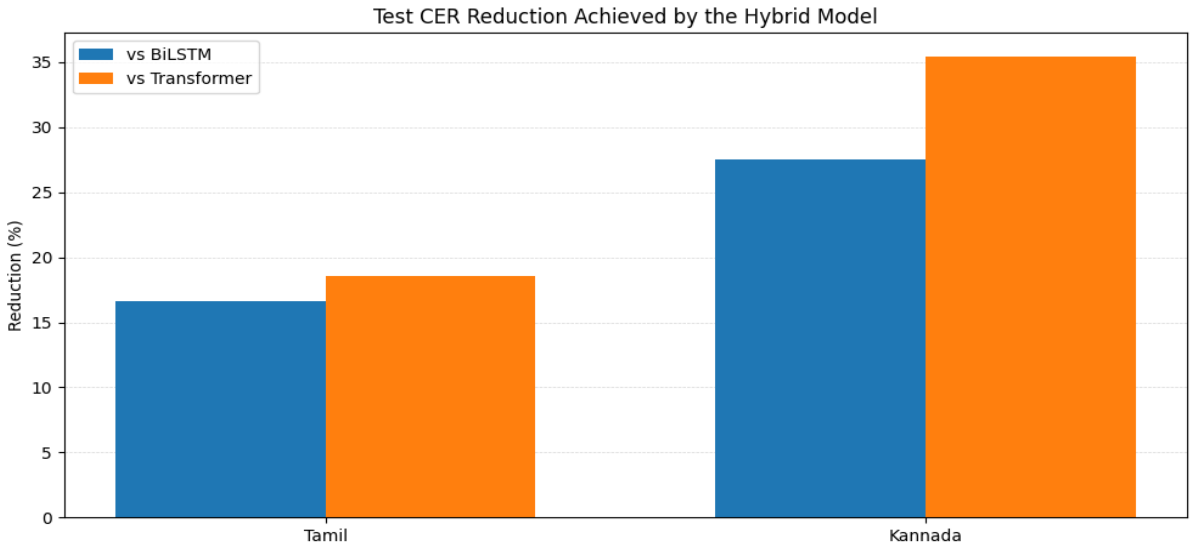

Figure 6. Test CER Reduction Achieved by the Hybrid Model.

Comparison of percentage CER reduction achieved by the hybrid model relative to the baseline architectures across Tamil and Kannada. Figures 5 and 6 together confirm that the proposed model improves performance consistently across both scripts. However, the magnitude of the gain is larger in Kannada, which suggests that script-level complexity influences the degree to which the hybrid architecture can leverage its combined spatial, structural, and contextual modeling capabilities.

4.3 Dataset-Specific Performance Analysis

This subsection analyses the performance of the proposed hybrid model individually between Tamil and Kannada. This kind of analysis is required since the two scripts are different in terms of structure composition, complexities in their graphemics and variations in the handwritings. They can be analysed separately to understand the difference in the performance of the model between the two datasets.

4.3.1 Tamil Dataset Performance

The Tamil data is distinguished by complicated grapheme structures, ligature-like unions and high variation of the handwriting. These aspects render Tamil handwritten word recognition very difficult. The overall number of test samples, correct predictions, incorrect predictions, test accuracy and test CER of the model on this script were calculated on the entire Tamil test set.

Table 2. Test Performance of the Hybrid Model on Tamil Dataset

Metric	Value
Total Test Samples	16184
Correct Predictions	13831
Wrong Predictions	2353
Test Accuracy	0.8546
Test CER	0.0281

Table 2 indicates that the model is accurate in predicting 13,831 out of 16,184 Tamil test samples, with high test accuracy of 85.46. The CER of 0.0281 means that despite the fact that the model is not accurate in the prediction of the complete word, its output can still be very close to the ground truth at the character level.

Along with aggregate performance, there is also a need to determine the change in recognition difficulty as a function of sequence length. Words with more than three letters tend to have more characters and cumulative decoding

mistakes. Word-level accuracy was calculated to explore this effect at various ground-truth word lengths in Tamil. The trend resulting is presented in Figure 7.

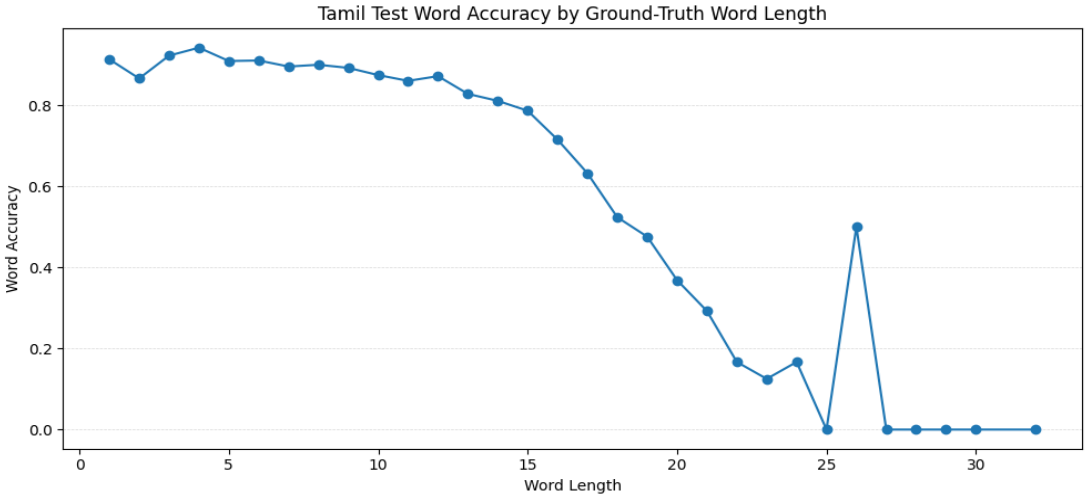

Figure 7. Tamil Test Word Accuracy by Ground-Truth Word Length.

Recognition accuracy of Tamil words at the word-level when comparing with ground-truths of varying word lengths. Figure 7 indicates that the model is effective with the shorter and medium length words in Tamil language, and the accuracy decreases with the increase in the word length. This is an indication of the fact that it is more difficult to acknowledge long hand written words in a script of complex structural composition.

4.3.2 Kannada Dataset Performance

In comparison to Tamil, Kannada handwriting in the present dataset seems to have a more established structural regularity, which could be easier to recognize. The identical assessment algorithm applied to Tamil was used to assess the Kannada test set to identify the overall performance of the dataset at the level of the whole dataset.

Table 3. Test Performance of the Hybrid Model on Kannada Dataset

Metric	**Value**
Total Test Samples	15730
Correct Predictions	13953
Wrong Predictions	1777
Test Accuracy	0.8870
Test CER	0.0175

Table 3 shows that the hybrid model identifies 13,953 of 15,730 samples of Kannada correctly. The increased word accuracy and reduced CER, relative to Tamil, both of which are attributable to the model, imply that the model is more capable of reflecting the structure of Kannada handwritten words. In order to test the sequence-length effects further, the word-level accuracy in Kannada was calculated at varying ground-truth word lengths. The pattern that results is presented in Figure 8.

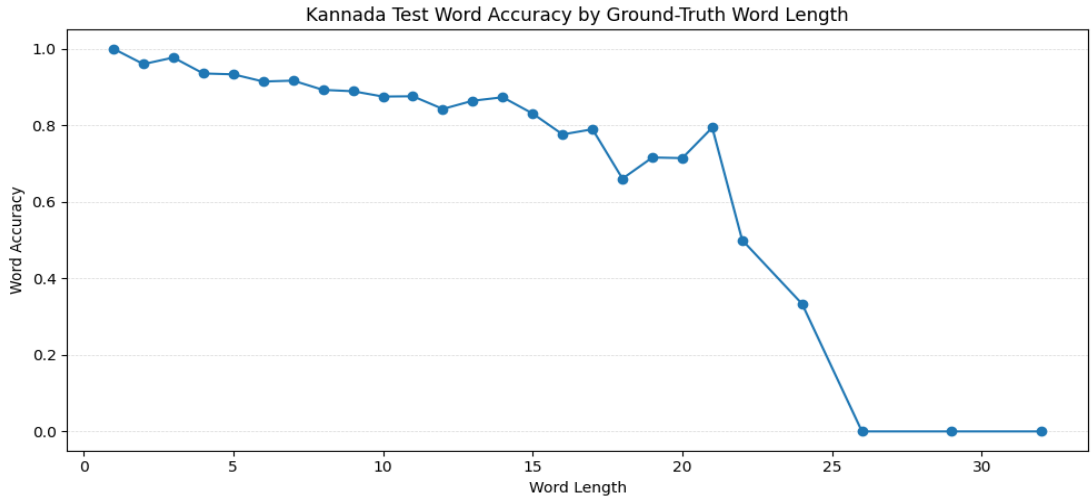

Figure 8. Kannada Test Word Accuracy by Ground-Truth Word Length.

Kannada Recognition accuracy of Kannada words at different ground-truth word lengths. In Figure 8, it is observed that the model is relatively stable in a wide range of Kannada word length. The decrease though still apparent with longer words, is less pronounced than that with Tamil which suggests greater sequence modeling stability. Figure 9 presents a direct comparison of the correct and incorrect predictions of the two datasets.

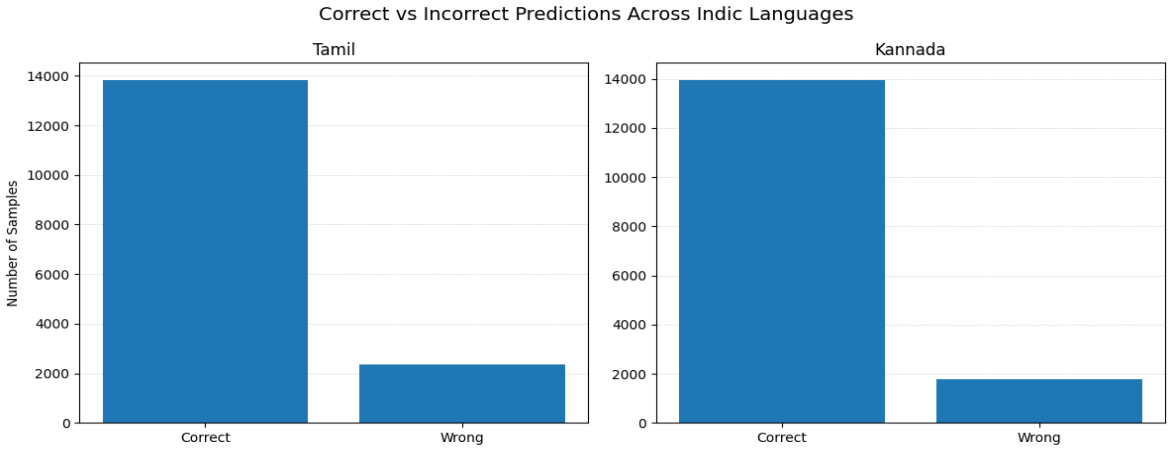

Figure 9. Correct vs Incorrect Predictions Across Indic Languages.

Comparison of the number of correct and incorrect predictions for Tamil and Kannada test sets.

Figure 9 reinforces the dataset-specific findings by showing that correct predictions substantially outnumber incorrect ones in both scripts, with Kannada showing a slightly stronger margin of correct recognition.

4.4 Error Analysis and Qualitative Evaluation

In handwritten text recognition, aggregate metrics such as accuracy and CER do not always reveal the nature of the underlying prediction errors. A model may produce an incorrect word prediction while still capturing most of the correct characters. Therefore, a deeper analysis at the sample level is necessary to understand whether the model fails completely or instead produces near-correct outputs.

4.4.1 Sample-Level Error Analysis

To analyze this behavior, representative ground-truth and predicted word pairs were examined along with correctness labels, word length, and edit distance. The edit distance provides a direct indication of how far an incorrect prediction is from the true word. The sample-level results are summarized in **Table 4**.

Table 4. Sample-Level Error Analysis of Model Predictions for Tamil and Kannada

Language	Ground Truth	Prediction	Correct	GT Length	Edit Distance
Tamil	ஆச்சரியமாய்ச்	அச்சரியமாய்க்	0	13	2
Tamil	கூட்டுகிறார்	கட்டுகிறார்	0	12	1
Tamil	இஞ்சினியரிங்	இஞ்சினியரிங்	1	12	0
Tamil	யூனியன்	யூனியன்	1	7	0
Tamil	வரப்போக	வரப்போக	1	7	0
Kannada	ಕಾಯ್ದಿರಿಸಿದ್ದ	ಕಾಯ್ದಿರಿಸಿದ್ದ	1	13	0
Kannada	ಸೋಲು	ಸೋಲು	1	4	0
Kannada	ಅದರರ್ಥ	ಅದರರ್ಥ	1	6	0
Kannada	ಹೊಂದಿ	ಹೊಂದಿ	1	5	0
Kannada	ಸ್ಥಾಪನೆಗೆ	ಸ್ಥಾಪನೆಗೆ	1	9	0

Table 4 shows that the majority of incorrect outputs, particularly in Tamil, are low-distance errors. This means that the model often captures the overall word structure correctly but makes a small number of character-level substitutions or omissions. Such behavior is preferable to complete misrecognition because it indicates that the learned representations remain strongly aligned with the target words.

4.4.2 Qualitative Prediction Analysis

To complement the tabular error statistics, qualitative examples were also examined from the validation and test predictions. These examples provide visual insight into how the model behaves on real handwritten inputs.

Figure 10. Qualitative Comparison: Validation vs Test Predictions (Tamil).

Representative Tamil handwritten samples with corresponding ground-truth and predicted outputs from validation and test sets. Figure 10 shows that the hybrid model is capable of preserving the overall lexical structure of Tamil words even in challenging handwritten forms. The qualitative evidence supports the earlier observation that most Tamil errors are localized rather than catastrophic. Representative Kannada examples are shown in **Figure 11**.

Figure 11. Qualitative Comparison: Validation vs Test Predictions (Kannada).

Representative Kannada handwritten samples with corresponding ground-truth and predicted outputs from validation and test sets. Figure 11 shows that Kannada predictions are generally highly accurate and closely aligned with the ground truth. This visual consistency agrees with the lower CER and higher accuracy reported earlier for Kannada.

4.4.3 Semantic Validation of Prediction Quality

Beyond exact character matching, it is also useful to assess whether the predicted outputs remain semantically meaningful. For this purpose, selected model predictions were inspected using external translation support to verify

whether the recognized words preserved their intended meaning. This semantic validation example is shown in **Figure 12**.

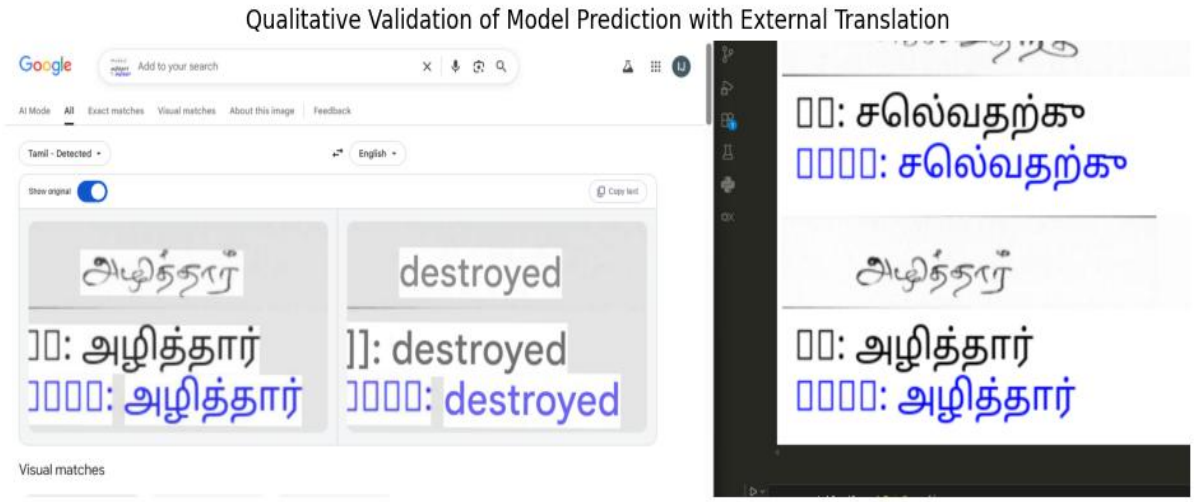

Figure 12. Qualitative Validation with External Translation Support.

Example of predicted handwritten text output cross-checked using external translation support to verify semantic plausibility. Figure 12 suggests that the predicted output remains semantically interpretable, further reinforcing the practical utility of the proposed recognition framework.

4.5 Cross-Script Evaluation Discussion

This subsection consolidates the findings across all experiments to assess the cross-script applicability of the proposed architecture. The results show that the same hybrid design consistently improves performance over both baselines for both Tamil and Kannada, even though the two scripts differ in complexity and visual structure.

The comparative tables demonstrate that the hybrid model achieves the best quantitative performance in both languages. The relative improvement figures show that the architecture produces not only higher accuracy but also substantially lower character-level error. The dataset-level analysis further indicates that the model maintains strong recognition quality in both settings, although longer and more complex Tamil words remain more challenging than Kannada.

Taken together, these findings demonstrate that the combined use of CNN, capsule, and Transformer modules offers a robust modeling strategy for Indic handwritten word recognition. The architecture appears to generalize well across scripts, suggesting that it can serve as a strong basis for future extensions to broader multi-script or multilingual handwritten text recognition tasks.

5. Discussion and Future Work

The findings show that the suggested Hybrid CNN Capsule Transformer architecture is beneficial as each of the elements performs a different and complementary role in the handwritten text recognition pipeline. The CNN element is the main visual encoder that is capable of extracting local stroke-level and texture-based features of handwritten word images. This is particularly crucial in Indic hand written recognition, where small variations in curvature, stroke continuity and character composition can greatly impact recognition quality. Previous image-based sequence recognition systems have also determined the advantage of robust visual feature extraction that precedes sequence decoding, especially in end-to-end recognition systems [3]. In the current work, CNN module is the representation one which offloads the subsequent structural and sequential modeling processes. The capsule component is a reinforcement of the architecture that maintains structural relations between extracted visual features. In contrast to the regular convolutional layers where the main focus is on the presence of features, the capsule mechanism is better adapted to preserving the compositional and part-whole relationships among the local features. This comes in handy especially in handwritten recognition of Indic words, where recognition errors are not only caused by the absence of features, but also by similarity of structurally similar components of characters. The capsule stage enhances representational expressiveness of the system by encoding more structural information between local

features and how they are structured. This is particularly useful in scripts where the formations of characters and modifiers may pose significant visual ambiguity, like in Tamil. The Transformer part plays its role in modeling long-range sequence dependencies of the encoded feature sequence. The visual classification task is not the only problem related to handwritten word recognition: it is also a sequential interpretation task, as the model has to predict the correct order of characters in a continuous image representation. The sequence modeling based on attention has been proven to offer good contextual learning in sequence tasks, as it is able to capture global dependencies more efficiently than its recurrent counterparts alone [7]. Transformer-based decoding and contextual refinement in handwritten text recognition in particular have been able to clearly achieve an increase in recognition accuracy as well [9]. The Transformer layer in the current architecture helps to supplement the CNN and capsule stages and make sure that the learned local and structural features are viewed in a larger order context. Effectiveness of the proposed model is hence the hybridization. Instead of basing the architecture on a single modeling mechanism, the architecture enjoys the complementary learning processes: CNN layers learn local discriminative features, capsule encoding learns structural relationships, and Transformer learns contextual dependencies. This combined architecture has a higher representation capacity compared to vanilla CNN-RNN or CNN-Transformer baselines as demonstrated in the form of higher word accuracy and reduced Character Error Rate on Tamil and Kannada datasets. The cross-script consistency of the improvements also suggests that the architecture is not just specialized to a single dataset but has a level of flexibility to scripts with a variety of structural properties. This is in line with recent trends in handwritten document recognition where more powerful hybrid and Transformer-based systems have further enhanced the quality of representation and recognition strength [12].

Although there are these strengths, the current research has a number of limitations. To start with, the model is not trained on all the scripts together, which implies that the existing framework is not an integrated multilingual or multi-script handwritten recognition model. Rather, it shows architectural applicability across the scripts by separate Tamil and Kannada experiments. Second, the experiment is confined to two Indic scripts and thus the applicability of the framework to a wider spectrum of the Indic systems of writing is yet to be explored. Third, despite the fact that the proposed model is more accurate in recognition, the efficiency of deployment, lightweight optimization, and real-time constraints of inferences are not explicitly considered in the current implementation, although they are critical issues in actual document processing applications. These findings can be important in the future work in a number of ways. A key future direction would be to create a real-life multilingual unified model that is trained together with various Indic scripts as opposed to training individual script-specific models. This kind of a framework may embrace common feature learning and script-aware adaptation in a single architecture. The other direction of importance is to expand the analysis to other Indic languages to allow analysis of the architectural robustness under more multi-script conditions. Application-wise, optimization of lightweight models and real-time deployment (in order to enhance usability in real-world recognition systems) should also be considered in the future. Lastly, language-aware or script-aware embeddings may also enhance the contextual disambiguation, especially with structurally similar character patterns within similar scripts.

6. Conclusion

This paper suggested a Hybrid CNN-Capsule-Transformer network to recognize Indic handwritten texts and tested on Tamil and Kannada handwritten text datasets. The architecture was also made in such a manner that it integrates local feature extraction, structural relationship modeling and long-range sequence dependency learning into a single framework. It was experimentally established that the proposed model outperformed the baseline architectures, with respect to word-level accuracy and Character Error rate on the two datasets. The cross-script analysis also revealed that the identical architectural design is effective in two structurally different Indic scripts. On the whole, the research confirms that the hybrid implementation of CNN, capsule, and Transformer modules offer a powerful and credible solution to Indic handwritten word recognition as well as it provides the basis to further developments with the wider multi-script and multi-language handwritten text recognition systems.

References

1. A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in *Proceedings of the 23rd international conference on Machine learning*, in ICML ’06. New York, NY, USA: Association for Computing Machinery, Jun. 2006, pp. 369–376. doi: 10.1145/1143844.1143891.
2. A. Graves and J. Schmidhuber, “Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks,” in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2008. Accessed: Apr. 16,2026.[Online].Available:https://proceedings.neurips.cc/paper/2008/hash/66368270ffd51418ec58bd793f2d9b1b-Abstract.html

3. B. Shi, X. Bai, and C. Yao, “An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition,” Jul. 21, 2015, *arXiv*: arXiv:1507.05717. doi: 10.48550/arXiv.1507.05717.
4. T. Bluche and R. Messina, “Gated Convolutional Recurrent Neural Networks for Multilingual Handwriting Recognition,” in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Kyoto: IEEE, Nov. 2017, pp. 646–651. doi: 10.1109/ICDAR.2017.111.
5. J. Puigcerver, “Are Multidimensional Recurrent Layers Really Necessary for Handwritten Text Recognition?,” in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Kyoto: IEEE, Nov. 2017, pp. 67–72. doi: 10.1109/ICDAR.2017.20.
6. M. Shafiq and Z. Gu, “Deep Residual Learning for Image Recognition: A Survey,” *Applied Sciences*, vol. 12, no. 18, p. 8972, Jan. 2022, doi: 10.3390/app12188972.
7. A. Vaswani *et al.*, “Attention is All you Need,” in *Advances in Neural Information Processing Systems*, Curran Associates,Inc.,2017.Accessed:Apr.16,2026.[Online].Available:https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html
8. S. Sabour, N. Frosst, and G. E. Hinton, “Dynamic Routing Between Capsules,” in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2017. Accessed: Apr. 16, 2026. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/hash/2cad8fa47bbef282badbb8de5374b894-Abstract.html
9. C. Wick, J. Zöllner, and T. Grüning, “Transformer for Handwritten Text Recognition Using Bidirectional Post-decoding,” in *Document Analysis and Recognition – ICDAR 2021*, J. Lladós, D. Lopresti, and S. Uchida, Eds., Cham: Springer International Publishing, 2021, pp. 112–126. doi: 10.1007/978-3-030-86334-0_8.
10. K. Barrere, Y. Soullard, A. Lemaitre, and B. Coüasnon, “A Light Transformer-Based Architecture for Handwritten Text Recognition,” in *Document Analysis Systems*, vol. 13237, S. Uchida, E. Barney, and V. Eglin, Eds., in Lecture Notes in Computer Science, vol. 13237. , Cham: Springer International Publishing, 2022, pp. 275–290. doi: 10.1007/978-3-031-06555-2_19.
11. M. Li *et al.*, “TrOCR: Transformer-Based Optical Character Recognition with Pre-trained Models,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 11, pp. 13094–13102, Jun. 2023, doi: 10.1609/aaai.v37i11.26538.
12. D. Parres, D. Anitei, and R. Paredes, “Handwritten Document Recognition Using Pre-trained Vision Transformers,” in *Document Analysis and Recognition - ICDAR 2024*, E. H. Barney Smith, M. Liwicki, and L. Peng, Eds., Cham: Springer Nature Switzerland, 2024, pp. 173–190. doi: 10.1007/978-3-031-70536-6_11.
13. G. Retsinas, K. Nikolaidou, and G. Sfikas, “Enhancing CRNN HTR Architectures with Transformer Blocks,” in *Document Analysis and Recognition - ICDAR 2024*, E. H. Barney Smith, M. Liwicki, and L. Peng, Eds., Cham: Springer Nature Switzerland, 2024, pp. 425–440. doi: 10.1007/978-3-031-70546-5_25.
14. Y. Li, D. Chen, T. Tang, and X. Shen, “HTR-VT: Handwritten text recognition with vision transformer,” *Pattern Recognition*, vol. 158, p. 110967, Feb. 2025, doi: 10.1016/j.patcog.2024.110967.
15. S. Gongidi and C. V. Jawahar, “iiit-indic-hw-words: A Dataset for Indic Handwritten Text Recognition,” in *Document Analysis and Recognition – ICDAR 2021*, J. Lladós, D. Lopresti, and S. Uchida, Eds., Cham: Springer International Publishing, 2021, pp. 444–459. doi: 10.1007/978-3-030-86337-1_30.
16. E. Lalitha, A. Mondal, and C. V. Jawahar, “Enhancing Accuracy in Indic Handwritten Text Recognition,” in *Computer Vision and Image Processing*, J. Kakarla, R. Balasubramanian, S. Murala, S. K. Vipparthi, and D. Gupta, Eds., Cham: Springer Nature Switzerland, 2026, pp. 234–248. doi: 10.1007/978-3-031-93688-3_17.
17. A. Mondal, K. Tulsyan, and C. V. Jawahar, “Bridging the Gap in Resource for Offline English Handwritten Text Recognition,” in *Document Analysis and Recognition - ICDAR 2024*, E. H. Barney Smith, M. Liwicki, and L. Peng, Eds., Cham: Springer Nature Switzerland, 2024, pp. 413–428. doi: 10.1007/978-3-031-70536-6_25.
18. K. Tulsyan, N. Srivastava, A. Mondal, and C. V. Jawahar, “A Benchmark System for Indian Language Text Recognition,” in *Document Analysis Systems*, X. Bai, D. Karatzas, and D. Lopresti, Eds., Cham: Springer International Publishing, 2020, pp. 74–88. doi: 10.1007/978-3-030-57058-3_6.
19. M. Dhiaf, A. C. Rouhou, Y. Kessentini, and S. B. Salem, “MSdocTr-Lite: A lite transformer for full page multi-script handwriting recognition,” *Pattern Recognition Letters*, vol. 169, pp. 28–34, May 2023, doi: 10.1016/j.patrec.2023.03.020.
20. S. Gongidi and C. V. Jawahar, “IIIT-INDIC-HW-WORDS: A dataset for Indic handwritten text recognition,” Centre for Visual Information Technology, IIIT Hyderabad, 2021. [Online]. Available: https://cvit.iiit.ac.in/research/projects/cvit-projects/iiit-indic-hw-words.
21. L. S. Anusha, Abhay Anandrao Deshpande, “Implementation of Pre-processing Techniques for Precise Classification of Ancient Kannada Epigraphs”, Engineering, Technology & Applied Science Research,Vol. 15, No. 3, pp-23592-235988, 2025.