Electronic Letters on Computer Vision and Image Analysis 24(2):81-103, 2025

A Systematic Framework for Sanskrit Character Recognition Using Deep Learning

Vrinda Kore[1], G. Dhruva[1], Sahana Rao[1], M. Vijitha[1] and P. Preethi[1]

[1] *Department of Computer Science and Engineering, PES University, India*

Received 25th of February, 2024; accepted 1st of May, 2025

Abstract

Sanskrit is widely acknowledged as one of the world's oldest surviving classical languages, yet its usage has continued to decline unabated in the present milieu. The erosion of its popularity is attributable to the absence of native speakers of Sanskrit and perceived inaccessibility to contemporary audiences. Notwithstanding, Sanskrit remains historically and culturally inseparable from the Indian subcontinent, with numerous religious manuscripts, epigraphical inscriptions, edicts and scientific literature written in the Devanagari script. Attempts to resuscitate the language have been largely unsuccessful as these attempts have relied extensively on laborious human transcription and translation. Such manual endeavors can be augmented with the use of efficient computational techniques to facilitate the transcription of voluminous manuscripts written in the Devanagari script. The emergence of deep learning frameworks has enabled researchers to overcome the drawbacks of conventional machine learning algorithms in developing efficient and extensible character recognition systems. Nevertheless, the advancement of character recognition frameworks varies across different Indic scripts.

In this context, this paper introduces an extensible framework for the transcription of handwritten Sanskrit manuscripts. In the absence of a benchmark dataset of handwritten Sanskrit characters, the authors introduce a comprehensive dataset to facilitate further downstream segmentation. The dataset, on augmentation, comprises over a hundred thousand samples and has been collected from over a hundred individuals. The paper explores an integrated approach to segmentation and accordingly delineates a systematic methodology for effectively segmenting Sanskrit words, incorporating techniques such as thresholding, zone-based classification, median bisection and projection profiles. The proposed technique accommodates a diverse array of characters and modifiers present in the Sanskrit script. Subsequently, a concurrent deep learning architecture parallelizes transcription using Neural Networks (CNN and Residual Networks). The deep learning models show accuracies exceeding 90%. This paper attempts to demonstrate the significance of systematic and structured approaches to machine transcription of low-resource languages.

Key Words: Sanskrit, Devanagari, low-resource languages, transcription, optical character recognition, segmentation, deep learning, neural networks

Correspondence to: <dhruva.pesu@gmail.com>

Recommended for acceptance by Angel D. Sappa
https://doi.org/10.5565/rev/elcvia.1850
ELCVIA ISSN:15108-5097
Published by Computer Vision Center / Universitat Autònoma de Barcelona, Barcelona, Spain

1 Introduction

Sanskrit is one of the oldest extant languages in India with an expansive and structured script and has been accorded a classical status in the Indian subcontinent. Its historical prominence manifests through its role as the administrative and colloquial language [1] across the Indian subcontinent for an estimated three millennia. A large family of Indo-Aryan scripts, such as Bangla, Bodo, Hindi, Marathi, Nepali, Odia, and Manipuri [2], are derived from the Devanagari script. Nevertheless, Sanskrit remains a vulnerable language, posing significant difficulties in bridging the chasm between Sanskrit and contemporary regional languages. Therefore, a systematic Sanskrit character recognition framework has the potential to extend its coverage to include these related languages. Such expansion would result in the development of a scalable character recognition framework that is generic, eliminating the necessity for language-specific frameworks. The transcription of Sanskrit manuscripts into digitally encoded character streams can facilitate the optimization of storage space utilization over larger image files. Such digitization emerges as a primordial step in conserving the multifaceted linguistic and cultural legacy inherent to the language and facilitates the promotion of Sanskrit as a link language to a cluster of derived Indo-Aryan languages in existence today. Digitization of manuscripts can assist linguists and researchers in indexing, searching, and translating [3] handwritten Sanskrit manuscripts. This provides an exciting opportunity for researchers to harness computational methods, to promote the accessibility of Sanskrit to a wider audience. Figure 1 is a modified representation of a language-agnostic character recognition framework.

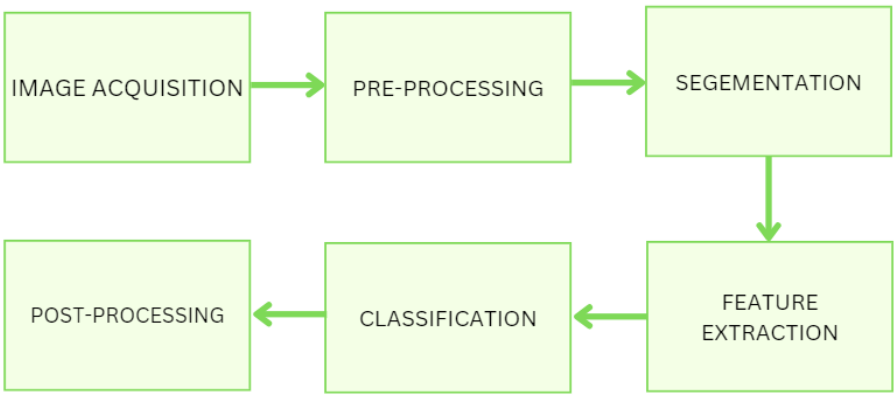

Figure 1: Language-Agnostic Character Recognition Framework

The progression of Optical Character Recognition (OCR) systems has transitioned from early template-matching techniques [4] to advanced deep learning architectures, driven by computational advancements and pattern recognition methodologies. Initial OCR models relied on predefined character templates, which proved inadequate for script variability. The introduction of probabilistic models, particularly Hidden Markov Models (HMMs) in the 1980s [5], enhanced OCR robustness by incorporating statistical dependencies, while feature extraction techniques such as zoning and contour analysis facilitated handwritten text recognition. The early 2000s saw the adoption of Principal Component Analysis (PCA) and clustering techniques for dimensionality reduction and text-line recognition. Contemporary OCR systems leverage deep learning architectures, including Convolutional Neural Networks (CNNs) [6], Residual Networks (ResNets), and Transformer-based models, which eliminate the need for manual feature extraction, enabling end-to-end recognition.

Sanskrit qualifies as a low-resource language primarily due to the paucity of digitized and annotated corpora requisite for training deep learning models, particularly for handwritten Devanagari script. While extensive Sanskrit texts exist, their computational accessibility remains constrained, with a significant proportion resid-

ing as non-digitized manuscripts, including texts with significant degradation. The inherent complexities of the script, including ligatures, conjuncts, and diacritic variability impede the use of transfer learning. Unlike high-resource languages with well-curated OCR benchmarks, Sanskrit lacks standardized repositories of handwritten characters, curtailing the efficacy of existing transcription frameworks. The unavailability of structured training data, the dearth of native speakers and the absence of domain-specific OCR architectures explain its classification as a low-resource language.

This paper proposes a systematic framework to assuage existing lacunae in the optical character recognition of Sanskrit. To overcome the initial barrier of a lack of handwritten datasets in Sanskrit, we introduce a benchmark dataset consisting of over a hundred thousand samples collected from over a hundred individuals. To facilitate further collaborative research in this domain, the dataset has been made available to scholars and researchers. Subsequently, the paper delineates a comprehensive methodology for the segmentation of handwritten words to decompose the given manuscript into its constituent components. In this context, this paper introduces a novel approach to compound character segmentation using median bisection. Subsequently, the paper outlines the use of a novel concurrent deep learning architecture to facilitate parallel transcription using Neural Networks. This paper presents a structured approach and includes sections that explore the complex nuances of the Sanskrit script, the extensive dataset collected, a downstream segmentation approach and concurrent deep learning architectures. This study incorporates compound character segmentation and classification, unlike existing research. With suitable modifications, the proposed methodology can be extended to a large cluster of derived languages, furthering the research on language-agnostic character recognition [7] techniques for derived Indo-Aryan scripts.

1.1 The Sanskrit Script

Despite a gradual decline in the number of Sanskrit speakers, the language retains a degree of popularity and continues to draw academic interest from scholars and enthusiasts. Across historical epochs, the Devanagari script has served as a prevalent writing system across numerous Indic scripts. Within the Sanskrit language, its characters are categorized into consonants vowels, modifiers (diacritics), compound characters, and conjuncts, presenting a significant challenge in developing a cohesive segmentation framework capable of encompassing this extensive array of characters. Sanskrit's enduring appeal lies in its meticulous grammatical structure, characterized by a diverse variety of modifiers, characters, ligatures and semantic intricacies. In the contemporary Sanskrit script, there are 13 vowels, 41 consonants (including occlusives and sonorants), and a multitude of compound characters and modifiers. Unlike Latin scripts, two consonants can fuse to form compound characters, making the number of baseline characters unusually long [8].

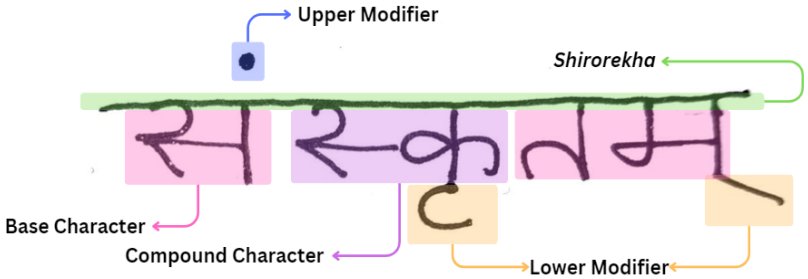

Figure 2: The Sanskrit Script

Constituent words in Sanskrit are strung together by a horizontal header line, called the Shirorekha [9]. This significantly hinders effective segmentation and subsequent isolation of the constituent characters for further classification. Yet, the Shirorekha is a defining feature of the Devanagari script and its usage is commonplace

in derived languages like Hindi, Gujarati and Bengali among others. Figure 2 shows an example that highlights the underlying nuances of the Sanskrit script. In stark comparison, the Latin script possesses a simpler structure. It primarily consists of letters representing individual phonemes, with limited diacritics used for accentuation and phonetic modifications. While Latin script offers straightforward phonetic representation, its versatility in representing complex linguistic constructs [10], especially those found in Sanskrit, is limited. Furthermore, Latin script lacks the intricate semantic rules that is characteristic of Devanagari.

1.2 Challenges

The endeavor to develop a structured and systematic framework for character recognition in Sanskrit presents a multi-pronged challenge, necessitating the elaboration of challenges before subsequent implementation. These challenges manifest in several steps, each posing distinct hurdles. Foremost among them is the absence of a comprehensive dataset tailored specifically for Sanskrit character recognition. The dearth of annotated data hampers the training process and compromises the accuracy and robustness of recognition systems. Furthermore, the inefficiency of existing architectures, an expansive range of characters and complexity in underlying ligatures present significant challenges. In this context, this section outlines each of the challenges.

The scarcity of annotated literary manuscripts [11] in Sanskrit has significantly impeded the development of scalable character recognition frameworks. Such a lacuna poses a significant obstacle to downstream computational advancements in transliteration and machine translation. Researchers continue to grapple with the absence of a standardized dataset that encompasses a comprehensive majority of Sanskrit characters, impeding any further progress in this field. Furthermore, existing datasets in Sanskrit comprise solely printed characters or contain minimal samples for each class. This poses a vexing challenge for researchers aiming to advance research in the domain of a language-agnostic framework for character recognition. Additionally, every new dataset requires several thousands of samples to be useful for training contemporary deep learning architectures. This problem is made more complex by the disparities in writing styles across people, as well as the inherent bias in writing and possible distortions in the way some characters are portrayed.

Although advancements have been made in the domain of character recognition of handwritten manuscripts, the proposed architectures cannot be replicated verbatim for low-resource languages like Sanskrit. Architectures involving sequential learning (such as LSTM) are constrictive, as they need a large corpus of annotated words [12]. Therefore, the proposed framework for low-resource languages must necessarily be accompanied by enhanced architecture and novel algorithms. Figure 3 is a high-level architecture of the Tesseract architecture [13] that is widely used for OCR frameworks across languages. The model's performance depends disproportionately on the availability of the aforementioned annotated word-level dataset. Furthermore, these architectures cannot be transferred to derived languages without extensive modification.

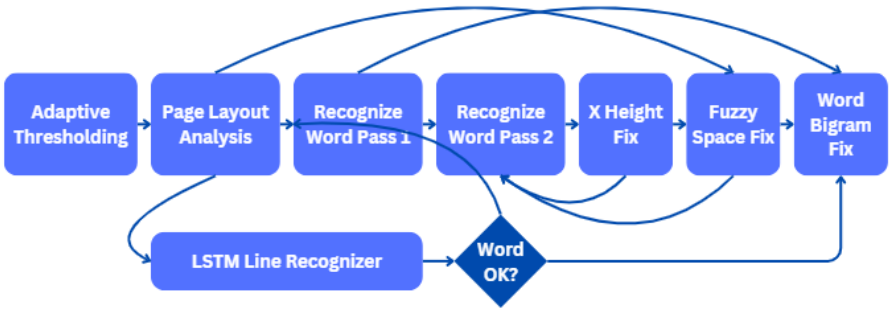

Figure 3: Tesseract Architecture using LSTM

Furthermore, twenty-six structurally separate and atomic characters in the English alphabet which are not concatenated further. By combining modifiers and compound characters, the Sanskrit script, on the other hand, comprises over a thousand distinct characters, resulting in greater diversity and flexibility in character combinations. The number of simple syllables is more than twice the number of English characters, with the exclusion of modifiers and complex characters. Moreover, the constituent characters and modifiers are connected by a horizontal header called the Shirorekha, which requires pre-processing to structurally decompose the given word into its base characters and modifiers. These challenges are peculiar to Devanagari-based scripts, requiring a far more nuanced and systematic approach [14] to character recognition with native algorithms and techniques.

2 Related Works

Avadesh et al. (2018) [15] propose a Convolutional Neural Network (CNN) based Optical Character Recognition system tailored for accurately digitizing ancient Sanskrit manuscripts. Several issues of the periodical *"Chandamama"* are utilized for constructing the dataset (as shown in Figure 4). The underlying dataset includes different font sizes and styles, enhancing the generalizability of the proposed OCR. The proposed methodology involves the use of image segmentation to identify letters, followed by training CNNs to classify these letters accurately. Furthermore, extensive pre-processing steps, including grayscale conversion, binary thresholding, line and word identification, header line removal, and compound character labeling are employed to enhance the quality and accuracy of the output. The test errors of the classifiers are reported as follows: Artificial Neural Network (ANN) has a test error of 21.79%, while Convnet A achieves 13.44%, Convnet B achieves 12.89 %, Convnet C achieves 10.41%, and Convnet D achieves the lowest test error of 6.68%. However, the study does not extend to handwritten character recognition.

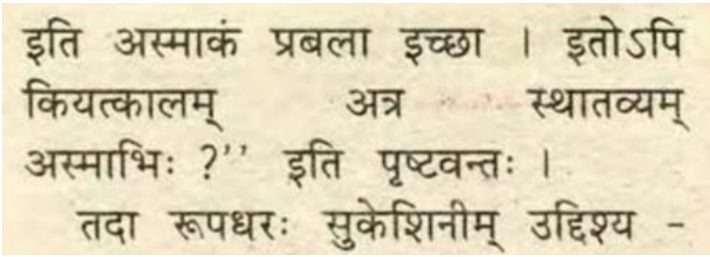

Figure 4: Training Data obtained from Sanskrit Periodicals

Halder & Roy (2011) [16] present a novel approach to effectively segment unconstrained handwritten Bangla words into distinct zones and individual characters. By leveraging histogram analysis and local segmentation techniques, the proposed method effectively identifies the "busy-zone", representing the region where a majority of characters reside, and subsequently partitions the word into upper, middle, and lower zones. Through a combination of horizontal projection and pixel-wise traversal, the method accurately isolates characters within each zone, mitigating issues such as overlapping and noise. Evaluation results demonstrate an accuracy of 81.41%, underscoring the efficacy of the proposed approach for segmentation of handwritten Bangla script. Figure 5 illustrates the proposed zone-wise segmentation, which serves as the basis for subsequent segmentation.

The methodology adopted by Gurav et al. (2020) [17] aims at enhancing Devanagari character recognition by integrating image processing and deep learning techniques. A specialized dataset is curated, comprising handwritten characters devoid of the Shirorekha. Image processing steps encompassed grayscale and binary conversion, Shirorekha detection via the Hough transform, and noise reduction. For transcription, deep learning

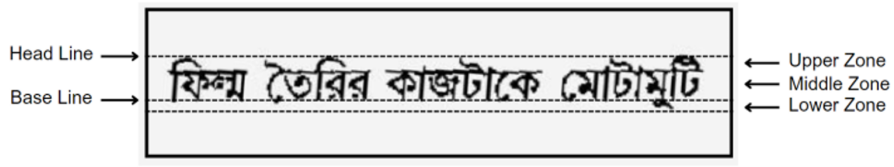

Figure 5: Zone-Based Segmentation

techniques were used with a Deep Convolutional Neural Network (DCNN) architecture featuring consecutive convolutional layers (as shown in Figure 6) for efficient high-level feature extraction. This DCNN approach enabled the system to attain an impressive accuracy, achieving 99.65% on training images and 98.22% on validation datasets. Despite some misclassifications between structurally similar characters, the proposed study underscores the efficacy of deep learning methodologies in advancing Devanagari character recognition over traditional Machine Learning techniques.

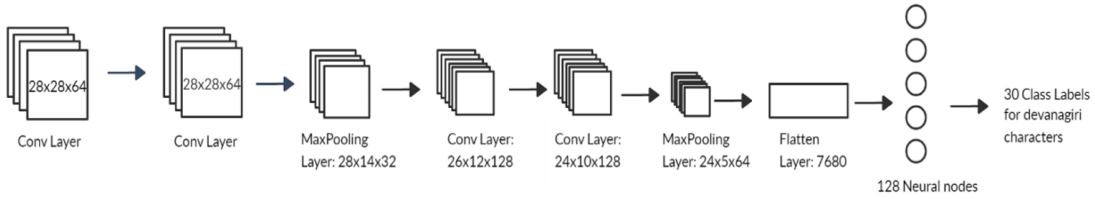

Figure 6: Proposed CNN Architecture

Kulkarineetham and Thananant (2023) [18] investigate the classification of Thai handwriting characters using a deep learning approach trained on the Burapha-TH dataset, which consists of 13,200 images across 44 Thai consonant classes. The authors employ Google Teachable Machine to develop a model trained on three dataset variations, each with increasing image counts per class (100, 200, and 300) (in Figure 7). Results indicate that larger datasets and increased training epochs lead to improved classification performance. The highest testing accuracy of 83.43% was achieved with the largest dataset (300 images per class) at 70 epochs and a learning rate of 0.0001. These findings highlight the impact of dataset size and hyperparameter tuning on handwriting character recognition accuracy, aligning with broader research emphasizing model optimization for handwritten script classification.

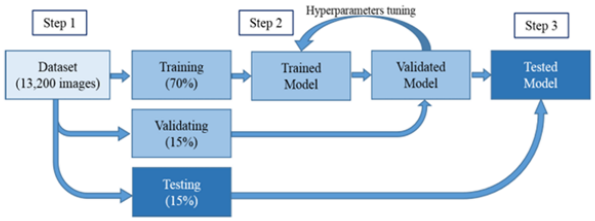

Figure 7: Proposed Model for Evaluating Performance on Handwritten Thai Characters

Li et al. (2018) [19] propose a two-stage convolutional neural network (CNN) combined with path signature features to improve rotation-free handwritten Chinese character recognition (HCCR). The methodology involves training a CNN to recognize both non-rotated and rotated characters (in Figure 8), predicting rotation angles, and correcting the character orientation before final classification. Network A was trained for both character recognition and rotation prediction, while Network B focused solely on character classification.

The results demonstrated that the proposed approach achieved 97.38% accuracy on the ICDAR-2013 HCCR dataset, significantly improving recognition for rotated characters while maintaining comparable performance to state-of-the-art methods for non-rotated characters.

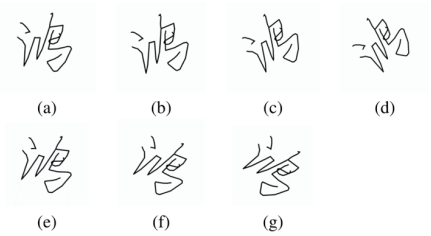

Figure 8: Rotation of a Handwritten Chinese Character

Mokhtar et al. (2018) [20] develop a sequence-to-sequence OCR correction framework using word-based and character-based RNN models. The word-based model employs a three-layer LSTM encoder-decoder with 1024 hidden units and dropout, while the character-based model adds a fourth layer and uses the Adam optimizer. They preprocess data through tokenization, dictionary construction, and normalization with Levenshtein distance. The framework is tested on English, German, and Medieval Latin datasets, showing that the character-based model consistently outperforms the word-based approach, particularly for unseen words. Results demonstrate higher accuracy compared to traditional correction methods, highlighting the effectiveness of deep learning in OCR error correction.

Yu (2021) [21] proposes an OCR-based translation guiding system that enhances handwritten character recognition using an online-offline combinational model. The method includes pre-processing scanned images, extracting stroke-based features, and applying clustering-based OCR for character classification. The recognized text integrates with a translation model using corpus technology and translation memory. The results show high recognition accuracy with low misclassification and rejection rates, demonstrating its effectiveness in handling variations. Despite challenges like noise interference, this approach significantly enhances OCR systems.

2.1 Theoretical Framework

The theoretical framework for Sanskrit character recognition using deep learning models encompasses several systematic steps. Primarily, the proposed framework requires data acquisition to build a robust dataset from a large number of individuals to mitigate potential bias. The dataset must undergo clustering [22], to ensure that similar samples are categorized together in corresponding buckets. Feature extraction techniques are used to capture the distinctive characteristics of samples, such as stroke patterns, curvatures, ligatures and junctions, and are used as the baseline for unsupervised clustering. On clustering, the data must be standardized using rigorous preprocessing techniques such as binarization, resizing, normalization, and augmentation to enhance the quality and diversity of the underlying training dataset. Due to the complex nature of the script, a structured segmentation framework must be used, to decompose several lines of handwritten text into atomic characters and modifiers for further classification. Subsequently, a concurrent deep learning architecture, using convolutional neural networks (CNNs) and Residual Networks (ResNet), is designed to learn hierarchical features and contextual dependencies crucial for accurate character classification. The concurrent architecture can parallelize the transcription of disparate zones, improving the efficacy of the framework. Lastly, comprehensive evaluation metrics, including accuracy and loss rates are utilized to benchmark the performance of the proposed models. This theoretical framework provides a systematic and structured approach for the development of a deep learning-based framework for Sanskrit character recognition. Figure 9 is a schematic representation of the proposed theoretical framework.

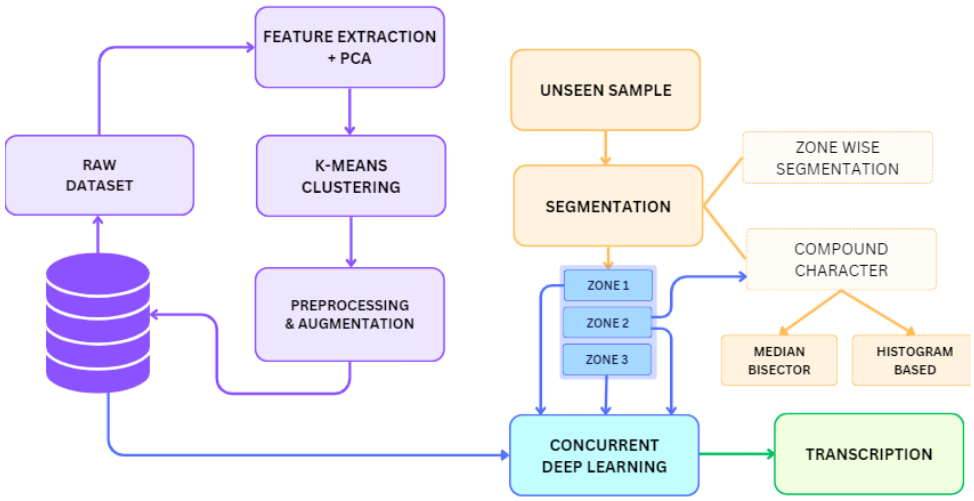

Figure 9: Proposed Theoretical Framework

The methodology proposed uses a comprehensive dataset obtained from a diverse pool of writers to train convolutional neural networks. Over 100 individuals were enlisted to create an unbiased corpus of handwritten Sanskrit characters, encompassing the varying styles of writing. This study introduces a novel downstream segmentation process, progressing from line level to word level and, subsequently, zone-based segmentation. This structured approach allows for a deeper understanding of the spatial relationships between characters, enhancing the model's capacity to accurately transcribe handwritten Sanskrit words. This paper introduces a novel median bisection-based technique for compound character segmentation within the identified zones. These techniques contribute to the overall effectiveness and generalizability of the proposed methodology. Finally, a novel concurrent deep learning architecture is applied to each segmented zone, facilitating the parallelized transcription of the handwritten Sanskrit words, while drastically reducing the number of classes.

The choice of this theoretical framework stems from the inherent complexities of handwritten Sanskrit recognition, particularly the script's extensive ligature formations, diacritic modifiers, and conjunct structures. A conventional OCR approach, which treats the script as a monolithic entity, struggles with the high degree of variability in handwritten forms, making a multi-stage, modular approach essential. The framework systematically decomposes the recognition process into discrete phases, each addressing a fundamental challenge unique to Sanskrit script processing. By structuring the framework around hierarchical segmentation and clustering, the model effectively isolates character components, enabling context-aware transcription. This theoretical formulation, based on pattern disambiguation and modular segmentation, optimizes feature extraction while preserving the inherent dependencies between Sanskrit graphemes.

3 Research Methodology

3.1 Determining Sufficient Number of Samples

The Power Theorem is applied to evaluate the statistical adequacy of the 16,000 unique samples collected for Sanskrit character recognition. Cohen's power analysis formula is used with appropriate assumptions.

$$n = \left(\frac{Z_{\alpha/2} + Z_{\beta}}{d} \right)^2 \approx 32$$

$Z_{\alpha/2} = 1.96$ (for a significance level of 0.05 (two-tailed test))
$Z_{\beta} = 0.84$ (for a power of 0.8)
$d = 0.5$ (moderate effect size)

This result indicates that 32 samples per class are necessary to observe non-trivial statistical effects. The number of samples for each of the 148 classes (approximately 111 samples) far exceeds this statistical sufficiency threshold required. However, the Vapnik-Chervonenkis (VC) theory of pattern recognition [23] has not been relied on. The VC theory quantifies a model's capacity to generalize by analyzing its VC dimension, which represents the largest set of points the model can shatter. We refrain from determining the correctness of the number of samples collected as Cohen's power analysis is used to determine the sample sufficiency rather than optimal class granularity. Therefore, the reliance placed on power analysis obviates our need to delve into the VC theory. To rigorously address class separability and learnability, techniques such as Structural Risk Minimization (SRM) under the VC theory would be required, which extend beyond the scope of the proposed research, and is left open for further research.

3.2 Framework for Dataset Collection

The absence of a benchmark dataset in Sanskrit poses a bottleneck [24] in the systematic design of a character recognition framework, making it imperative to curate an expansive dataset with a wide coverage of characters in the Devanagari script. In a departure from existing datasets, the proposed dataset consists of atomic words, half characters and modifiers (as opposed to a word-level dataset). This facilitates further segmentation and character recognition with a drastically reduced number of classes, as discussed in the subsequent sub-sections. Each of the base characters and modifiers can be combined to form a wide range of characters with different phonetic and literary significance.

The data acquisition process involved a diverse cohort of participants and employed a standardized grid-box approach. Participants were asked to provide sequential samples for isolated characters within the grid-box [25]. The aforementioned set of characters was vetted by a Sanskrit scholar to ensure the veracity of the dataset. Care was taken to ensure that the given samples were written inside the grids to mitigate the occurrence of characters exceeding designated boundaries, thus enhancing the completeness of the samples. The dataset comprised a large array of characters and modifiers, encompassing 11 vowels, 21 modifiers, 41 consonants, 48 recurring compound characters, and 27 half characters. Notably, vowels, consonants, and compound characters were recorded on an 11x11 grid, while the secondary sheet featured a 3x9 grid dedicated to half characters. Further details on the specificities of these entries are furnished in subsequent sections.

The corpus of the collected samples facilitates the derivation of a large number of Sanskrit characters, either as atomic components within the grid or as combinations of characters/modifiers present therein. Furthermore, demographic data was solicited from participants, by collecting attributes such as age, gender, and proficiency in utilizing the Sanskrit/Devanagari script. This serves as a critical metric in assessing potential biases inherent within the proposed dataset, thereby ensuring its scientific rigor and applicability.

3.3 Participants

Participants from across Karnataka, India contributed to the benchmark dataset. Adequate care was taken to ensure diversity among the participants (across age, gender and experience in the Devanagari script), to mitigate potential bias [26] in the dataset. The participants were asked to provide samples sequentially using a pen. An instance of a filled grid-box is shown above. 111 participants contributed to the dataset, resulting in a corpus of over 16,000 isolated samples after preliminary data cleaning [27]. The demographic diversity of the participants has been recorded in Table 1.

Table 1: Participant Demographic

Sl No	Age Demographic	Gender		Count
		Female	*Male*	
1	$<= 15$	14	3	17
2	$16 - 30$	48	28	77
3	$31 - 45$	4	9	14
4	$>= 45$	4	1	5
	Total	70	41	111

3.4 Pre-Processing, Feature Extraction & Clustering

After the data was collated, samples were scanned and archived for further analysis & pre-processing. The data was cropped recursively using the grids as a reference and placed in a centralized folder. In order to build a repository of isolated characters, this procedure was carried out for each of the datasets collected. Here, missing grids or incorrectly written samples are summarily removed to enhance the accuracy of the underlying deep-learning models. Inasmuch as this method may lead to a reduction in the number of samples for specific classes, they are compensated for in subsequent stages through data augmentation, thereby resulting in a balanced dataset.

The initial pre-processing of the data employs standard image processing methodologies such as grayscale conversion, binarization, and resizing. Throughout each stage of pre-processing, manual inspection is again conducted to identify and eliminate noisy outliers, thereby preempting potential cascading issues. Grayscaling standardizes samples by eliminating color variations stemming from the use of different ink colors by different participants, resulting in uniform grey images. Due to manual cropping adjustments made to rectify improper grid removal in certain character samples, these images lack standardized dimensions. To skewed proportions, images are resized to 64x64 pixels, facilitating subsequent feature extraction & clustering.

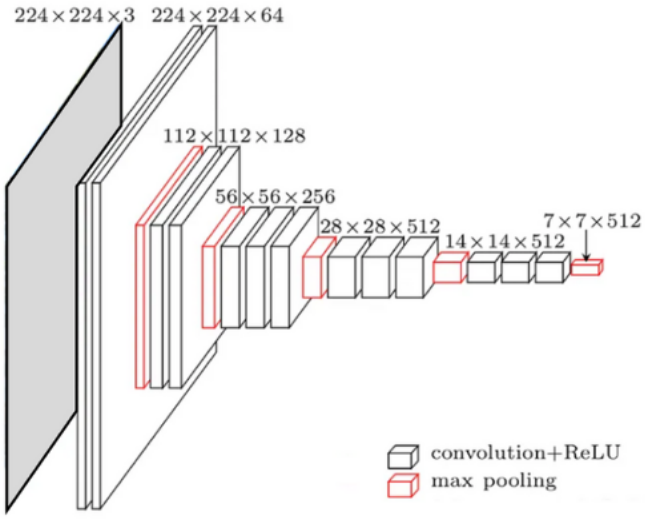

Figure 10: Modified VGG-16 Architecture for Feature Extraction

The samples are now standardized and stored in a single corpus for further categorization into corresponding bins using unsupervised clustering. As the samples are images, they cannot be clustered directly [28], and require further decomposition into vectors. K-Means clustering is used as a means to segregate samples into their respective bins with minimal effort. Despite the arduous nature of the process and the potential for a large number of samples to get misclassified, these errors are minimized through the adjustment of parameters and the implementation of robust feature extraction algorithms. Moreover, manual clustering of misclassified images into their respective bins is a foregone necessity. The methodology and the accuracy of clustering are extensively discussed below.

In this paper, we employ a pre-trained machine-learning model for feature extraction to facilitate further clustering. The VGG-16 architecture (schematically represented in Figure 10), known for its transfer learning capabilities, is used for this purpose by extracting features from the input layer to the last max pooling layer (with output dimensions 7 x 7 x 512), as depicted in Figure 10. The model yields a set of feature vectors corresponding to the input data, representing the transformation of images into a set of sparse vectors. These feature vectors are then consolidated into an array and reshaped to facilitate dimensionality reduction through Principal Component Analysis (PCA) [29].

Because of its large dimensionality, the VGG-16 model produces a highly sparse matrix with 4096 features, making it unsuitable for clustering. Nevertheless, we can retain the image's key attributes and features by utilizing Principal Component Analysis (PCA), which also facilitates the drastic reduction in the number of dimensions to 1551 features while keeping 95% of the variance. In addition to enhancing clustering performance, PCA identifies the most relevant features while minimizing the impact of noise or irrelevant features.

An unsupervised machine learning technique called K-Means clustering [30] is utilized to partition the given datasets into discrete groups, or clusters, according to the similarities between data points. On feature extraction, the vectors are given as input to the K-Means clustering algorithm, which partitions the singular corpus of data into 148 bins (as there are 148 unique characters in the proposed dataset). However, the result of the clustering is not uniform, as evidenced by the disparity in bin sizes shown in Figure 11.

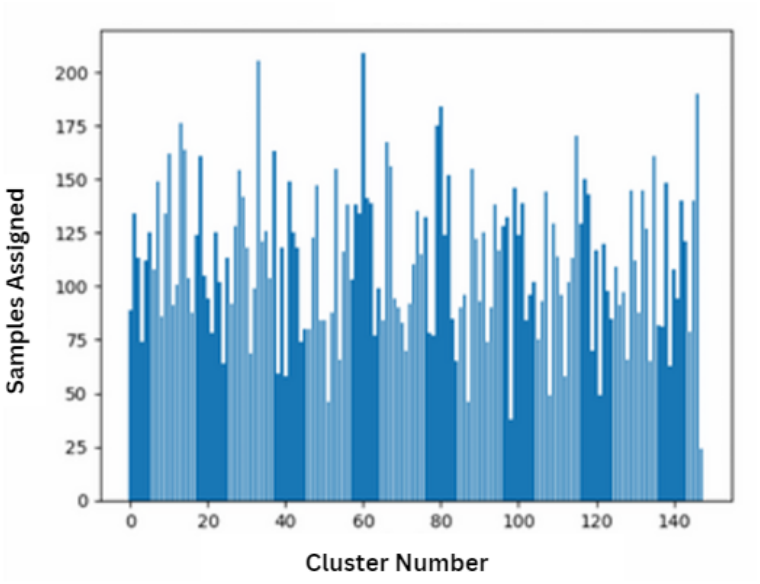

Figure 11: Misclassification after K-Means Clustering

The sizes of the clusters vary; the smallest cluster has 24 samples, while the largest cluster has 209 samples. The clustering technique correctly categorised a number of simple characters, as anticipated. Characters that are often used as terminals, such as || and | have high true positive rates of 98.19% and 100%, respectively. However, the accuracy differed based on the type of character under consideration. The accuracy rate for the virāma/halanta was 63.96%, but the vowel 'a' had a substantially lower rate of 18.01%. This disparity might be explained by the visual resemblance to other characters like 'u' and 'ū'. Compound characters performed comparatively poorly due to their complex writing syntax. The character 'tta', for instance, has the lowest true positive rate (15.31%) in the corpus as a whole. Furthermore, similar compound characters were often grouped together, such as 'sta', 'sma' and 'ska'. The performance of the K-Means clustering algorithm can be calculated by evaluating the percentage of samples that are classified correctly. Accordingly, it has been tabulated in Table 2.

Table 2: True Positivity Rates for Clustering

Sl No.	Type	Lowest TPR	Highest TPR
1	Simple Characters	18%	100%
2	Compound Characters	15.31%	66.67%
3	Half Characters	21%	75.68%
4	Modifiers	19.82%	91.89%

With over 16,000 unique samples, the dataset is further refined by using image processing techniques to augment the given data. This is mainly performed to prevent the deep learning models from overfitting and thereby increases their generalization. The methods used for augmentation include skewing (by an angle of 20°), thinning, and thickening. Subsequently, an augmented corpus of over 100,000 images is obtained, making it appropriate for training deep learning models.

3.5 Segmentation

Due to the intrinsic intricacy of the Sanskrit script and the difficulties mentioned in earlier sections, it is impractical to rely on a single, cohesive segmentation framework [31]. Therefore, to effectively segment the numerous components in a given manuscript, a step-by-step framework must be introduced, highlighting the need for a systematic algorithm for segmentation. A downstream segmentation framework, which attempts to efficiently subdivide a given manuscript into its component characters and modifiers, can aid in the development of a scalable character recognition framework for Devanagari and its derived scripts. First, the given lines of handwritten words are segmented into words and then zone-based slicing is applied to ensure accurate character segmentation. Notably, most Sanskrit letters and modifiers can be accommodated by this paradigm, highlighting superior generalization. Figure 12 is a representation of the proposed segmentation framework with corresponding algorithms.

In the first stage of the multi-step segmentation procedure, a given text document is broken down into its individual lines, including the modifiers and the Shirorekha [32]. The recommended method for line-level segmentation involves thresholding to make the lines more distinguishable, dilatation to bridge unconnected line segments, and contour extraction to define each line. The given manuscript is transformed into grayscale in order to standardize it. The image is subsequently subjected to a thresholding procedure, where an arbitrary threshold value of 80 is selected by means of iterative testing. The input is binarized as the underlying spatial structure supersedes the importance of color variations. To ensure continuity and bridge gaps between disconnected line segments in the binarized image, dilation is employed.

Defined by a kernel, each pixel's value is substituted with the local maxima from a neighborhood to dilate the image. This is accomplished by using a 7x118 rectangular kernel, which produces a well-connected region

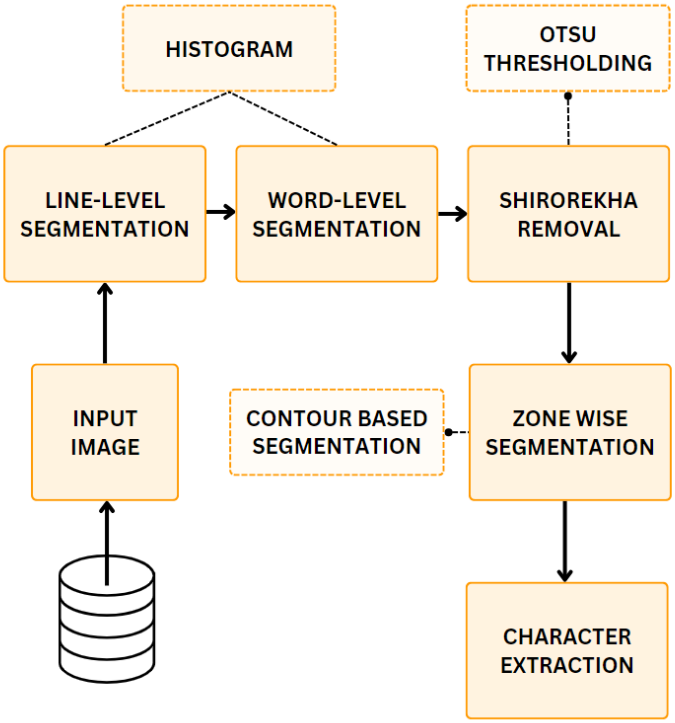

Figure 12: Downstream Segmentation Framework

that represents each line. Then, using a contour extraction approach, the contours are retrieved from the dilated picture, highlighting all related areas in the binarized image. Every contour has a bounding box assigned to it, and contours are arranged according to their y-coordinates to maintain the implicit text line order. Following line segmentation, words are segmented using analogous techniques. Contours are identified within each line, and words are sorted based on their x-coordinates to maintain their implicit order. The contour area serves as a parameter to filter out extraneous noise in the image. Figure 13 showcases the results obtained after line & word-level segmentation, where each segmented region is extracted and saved as individual images for further processing steps.

The Shirorekha acts as a horizontal delimiter, binding characters together to form a word. Bhat et al. (2020) [33] use statistical techniques to demonstrate that the Shirorekha exists in almost 99% of Devanagari manuscripts, making it an essential characteristic of the script. Furthermore, the Shirorekha is used for line-based segmentation, script recognition, and skew detection and correction. Therefore, it becomes necessary to include this step in the overall segmentation procedure. The removal of the Shirorekha is essential for further segmentation [34]. The first step in improving the detection of the Shirorekha is to reduce noise by using a Gaussian blur technique. Subsequently, the image's grayscale representation is transformed into a binary form using an arbitrary threshold, and Otsu's thresholding approach is applied for superior line detection. This method combines pixel intensity analysis, thresholding, and filtering to highlight and visualize important structural components in the image, with a focus on isolating the Shirorekha. In this context, Otsu's method is utilized to optimize this threshold by minimizing the intra-class variance of pixel intensities within the resulting binary image.

The binary picture is then used to create a vertical histogram by averaging the pixel intensities along the vertical axis. In this procedure, threshold values — arbitrary terms such as threshold high and threshold low —

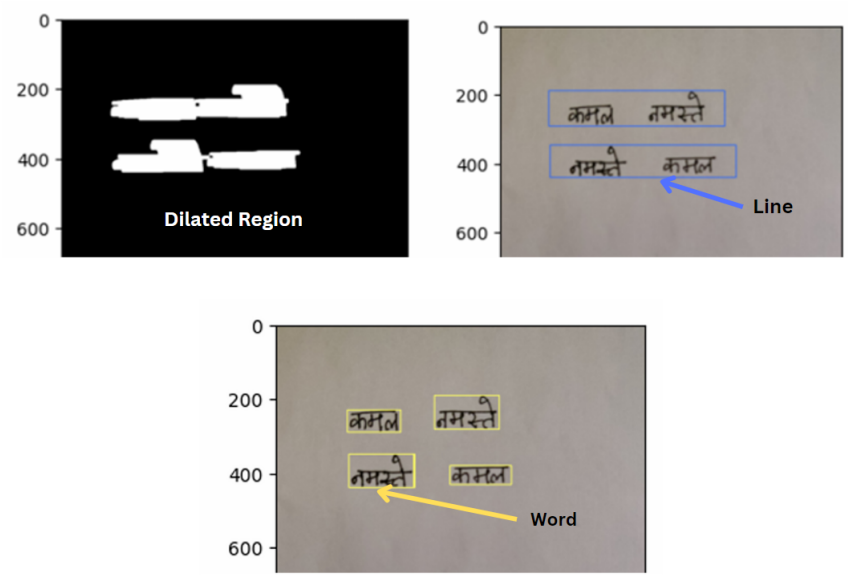

Figure 13: Line & Word-Level Segmentation

are used. Then, possible upper bounds of lines are found by repeatedly examining vertical locations to make sure that the average intensity at a particular location is less than a high threshold but greater than it is at the subsequent location. The pixels that make up the Shirorekha are efficiently grouped by this iterative procedure. The characters and modifiers are then effectively isolated from the Shirorekha by insetting a horizontal white line in place of the indicated pixels (as depicted in Figure 14). This becomes a conduit for further segmentation, as the characters and modifiers are effectively isolated by the removal of the Shirorekha.

Figure 14: Shirorekha Removal

Zone-based segmentation is an effective technique used in character recognition frameworks for Devanagari-based languages to effectively segment characters from a word. Using a zone-wise decomposition methodology results in considerable performance increase [35] over conventional segmentation techniques, especially for Indic scripts like Devanagari. This method significantly lowers the number of classes needed in later stages when deep-learning techniques are introduced. The zone-based technique involves the discrimination of three distinct sections within a given Sanskrit word, namely the upper zone, middle zone, and lower zone. The upper zone primarily encompasses modifiers (e.g., 'e' vowel sign and 'anusvara'), whereas the middle zone comprises full characters and compound characters (e.g., 'ka' and 'kha'). On the other hand, the lower zone contains additional modifiers (e.g., 'u' (vowel sign) and the halant ('virama')). This proposed segmentation strategy aids in effectively delineating the various components of Sanskrit words for further recognition [36].

The preliminary step involves configuring the parameters for the kernel size and the number of iterations for morphological closing. Subsequently, morphological closing is applied to a binary image to facilitate improved character grouping. Contours outlining the characters are then identified, and based on their vertical positions,

they are categorized as upper, lower, or middle characters. Visually discernible bounding boxes are delineated around the characters (as shown in Figure 15), and individual character images are extracted, denoting their respective zones (upper, lower, middle) and associated positions. These bounding boxes are colored red, blue and green respectively, and individual segments are cropped and stored for further classification. The coordinates of the segments are used to ensure in-order transcription of the three zones by three concurrent deep learning models.

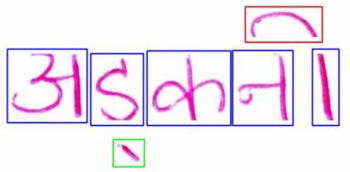

Figure 15: Zone-wise Segmentation

3.6 Transcription using Deep Learning

The aforementioned segmentation process is a conduit for further transcription using deep learning. Deep learning models are architected in parallel for each of the zones (as shown in Figure 16). This enhances the accuracy of the models while drastically reducing the number of classes that need to be trained. The classifications from each of the three zones are collated and the resultant output represents the transcription of the given handwritten sample. Each segmented zone is concurrently passed to its dedicated deep learning layer. This parallelization allows for concurrent transcription of the distinct components within the Sanskrit script, optimizing the model's ability to discern details present in upper modifiers, middle characters, and lower modifiers. By employing separate deep learning layers for each zone, the architecture ensures a nuanced and more accurate classification process.

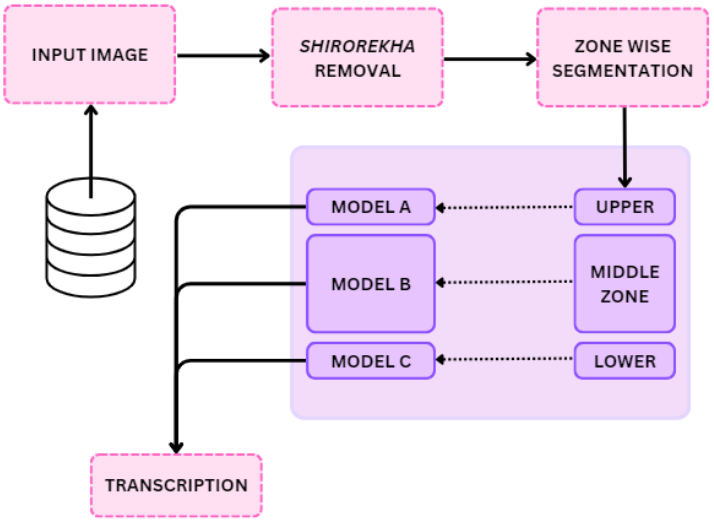

Figure 16: Concurrent Deep Learning Architecture for Character Recognition

Various Convolutional Networks and Residual Networks were experimented for each of the zones. The best validation results were obtained on simple CNN models [37]. The CNN layers for each of the zones

have been tailored after extensive experimentation. We employed a pre-trained ResNet model (Inception V2) [38] Despite significantly good training accuracy, the validation accuracy remained unsatisfactory, indicating significant overfitting. Furthermore, the training was computationally expensive on a T4 GPU accelerated environment. A significant drawback of using pre-trained models (such as ResNet V2) is the limited flexibility in experimenting with the layers. To mitigate this, we resorted to baseline CNN layers which were modified for each of the zones after experimentation. Additionally, the number of epochs was tweaked based on the number of classes in the zone. The input images are initially rescaled, followed by three convolutional layers with varying filter sizes and numbers, interspersed with max pooling and dropout layers to enhance feature extraction and mitigate overfitting. The last convolutional layer is followed by flattening the output into a one-dimensional vector. Subsequently, the flattened vector is fed into a series of densely connected layers (Dense), forming a multi-layer perceptron (MLP) architecture. The results (in Table 3) have been discussed below with appropriate graphs for CNN models (in Figure 17).

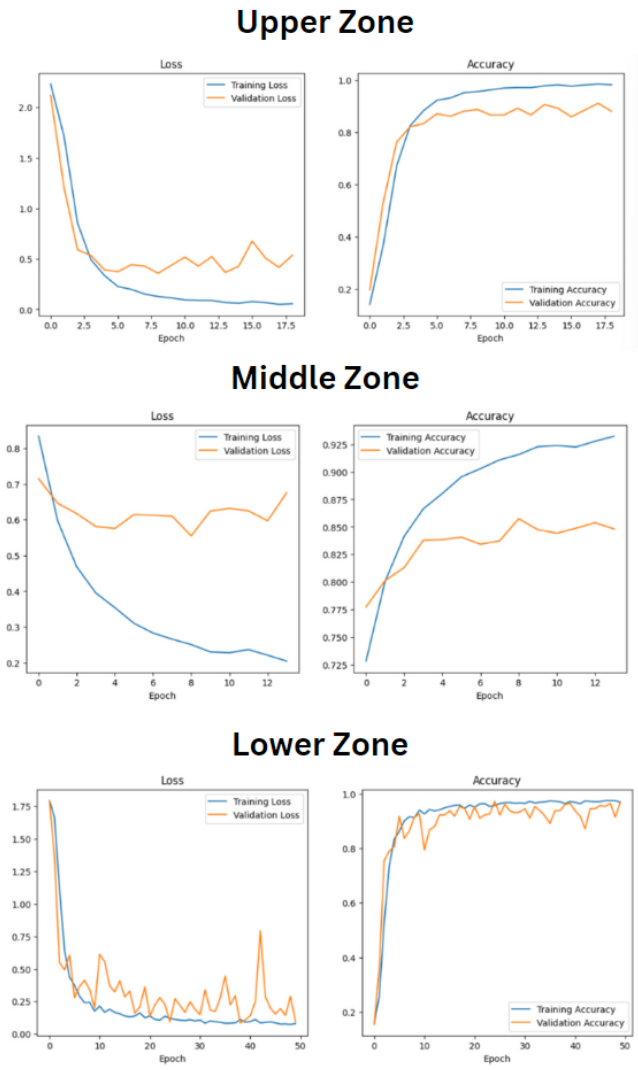

Figure 17: Graphs for CNN

Table 3: Results of CNN & ResNet V2

Sl No.	Zone	CNN Accuracy	Loss	ResNet V2 Accuracy	Loss
1	Upper Zone	98.16%	0.574	98.18%	0.541
2	Middle Zone	93.22%	0.698	94.01%	1.839
3	Lower Zone	96.97%	0.798	94.86%	1.596
4	Half Characters	95.11%	0.125	95.60%	1.370

3.7 Transcription using Spatial Referencing

To facilitate the faithful transcription of long vowels and short vowels, we introduce a spatial referencing-based technique to enhance the accuracy of the proposed model. The relative position of the modifier in the middle zone is compared with the position of the corresponding base character and modifier in the upper zone. Accordingly, the modifier is identified to be either a long or a short vowel. This uses the inherent structural properties of Devanagari script, where vowel modifiers maintain consistent positional patterns relative to their base characters.

The proposed technique is used to differentiate between a wide range of modifiers such as short 'i' versus long 'ī' vowel signs. The shirorekha or headline, serves as a critical anchor point and is used as the reference for assigning the appropriate upper modifier, as shown in Figure 18. By establishing this vertical reference line, the model can accurately determine whether a vowel modifier appears to the left or right of the baseline character, thereby distinguishing between visually similar diacritical marks that would otherwise be cumbersome to classify independently.

Such a technique aids the reduction in the number of classes on which the model is trained, thereby improving computational efficiency and reducing the complexity of the classification task. Instead of training separate classes for every possible character-modifier combination, the model maps base characters and modifiers independently, then applies spatial reasoning to reconstruct the complete character. This modular approach not only decreases training time but also enhances the model's ability to generalize to unseen character combinations.

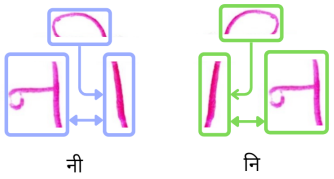

Figure 18: Transcription using Spatial Referencing

3.8 Compound Character Transcription

A compound character is formed by the fusion of a half-character with a full character. The lacuna of research for compound character transcription makes the development of a novel technique imperative. Constructing a dataset to encompass every possible compound character is infeasible and redundant [39]. To address the intricacies of compound characters within the middle zone, additional refinement is incorporated (as shown in Figure 19) into the proposed deep-learning architecture.

After the parallelized classification, if the confidence level of a recognized compound character in the middle zone falls below a predetermined threshold (80%), we dynamically implement a compound character segmentation strategy. In such instances, the image is bisected along its width, with the first half directed to a

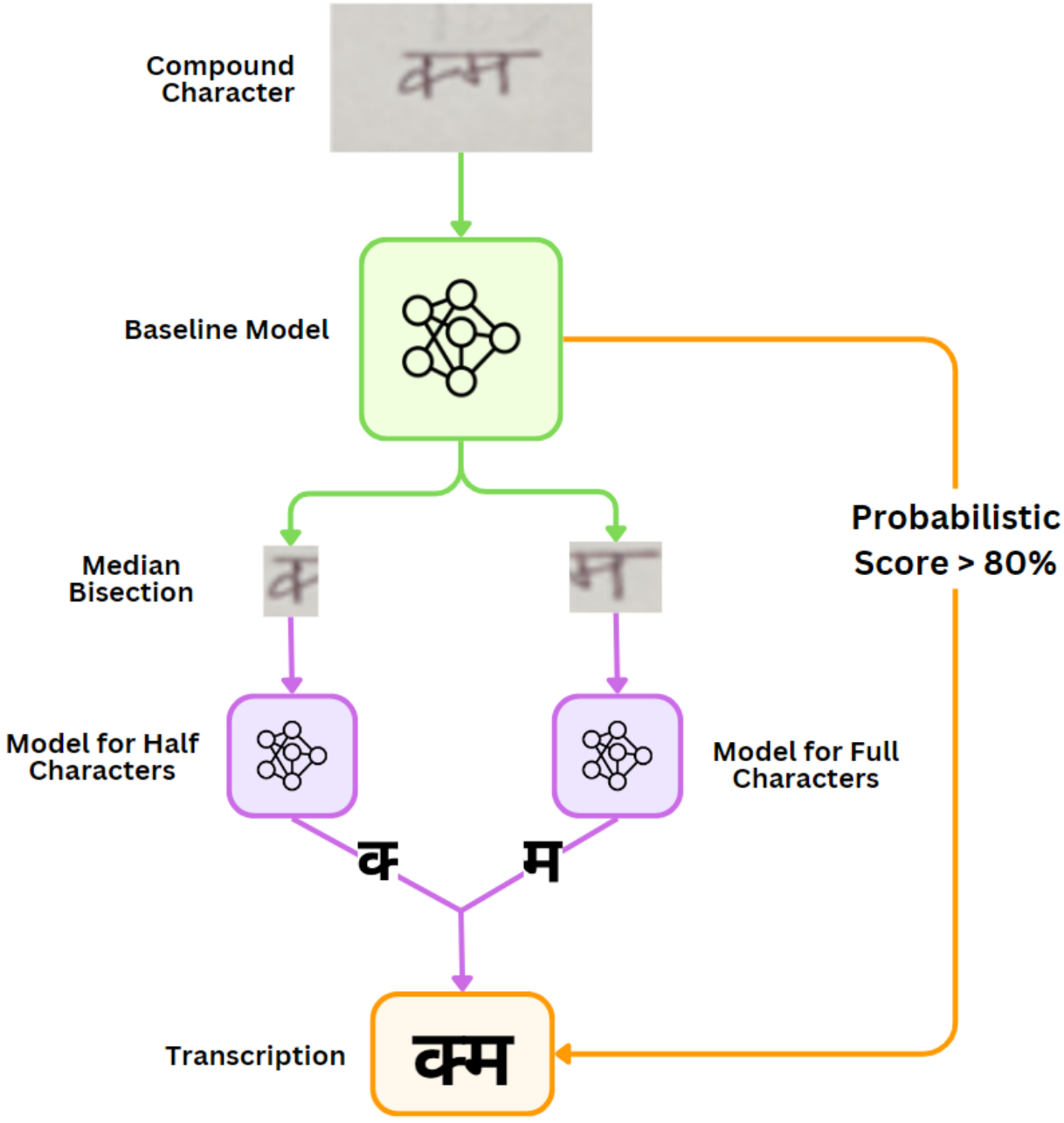

Figure 19: Compound Character Transcription Pathway

specialized half-character classifier, and the second half is passed to a comprehensive full-model classifier. The outputs from each of the layers are combined to produce the compound character, although the model has never been trained to recognize the compound character. The proposed strategy offers coverage of nearly 78% of compound characters (1299 compound character combinations in all), drastically reducing the number of compound character variants that the model needs to be trained on.

4 Conclusion & Future Work

The proposed framework describes a systematic methodology for Sanskrit character recognition by integrating structured segmentation techniques with deep learning architectures tailored for Devanagari-based scripts. The multi-stage preprocessing pipeline mitigates the intrinsic complexities associated with ligatures, diacritic modifiers, and conjunct formations. The use of a concurrent deep-learning architecture and leveraging Convolutional Neural Networks (CNNs) for spatial feature extraction, ensures efficient transcription of handwritten Sanskrit characters with minimal misclassification. The integration of spatial referencing in modifier classification and compound character resolution, augments the framework's efficacy in handling script variations based on semantic rules. The evaluations outline the superiority of zone-based classification over monolithic transcription, demonstrating significant improvements in recognition accuracy.

The empirical validation of the proposed methodology demonstrates substantial improvements over conventional OCR approaches, with the concurrent architecture achieving 98.16% accuracy for upper zone modifiers, 93.22% for middle zone characters, and 96.97% for lower zone diacritics. The novel compound character segmentation strategy, employing confidence-threshold-based median bisection handles compound character combinations without explicit training on complete conjunct forms. This significantly reduces computational overhead while maintaining high classification fidelity across morphologically complex character classes. The benchmark dataset, comprising 16,000+ unique samples from diverse participants, establishes a reproducible foundation for future research. Furthermore, the framework's language-agnostic design principles, particularly the zone-based decomposition and spatial referencing mechanisms, demonstrate transferability to cognate Indic scripts such as Hindi, Marathi, and Nepali, thereby addressing the broader challenge of low-resource language digitization within the Devanagari family.

Despite the demonstrated efficacy of the proposed methodology, several avenues warrant further exploration to enhance system performance and generalizability. Researchers can focus on refining the segmentation framework by incorporating reinforcement learning-based adaptive thresholding techniques, enabling dynamic parameter tuning for improved character isolation in noisy manuscript images. Additionally, the integration of transformer-based self-attention architectures, such as Vision Transformers (ViTs), can be explored to capture long-range dependencies within Devanagari character sequences, reducing error propagation in complex ligature formations. Given the limitations of conventional OCR in handling rare Sanskrit glyphs and stylistic variations, the implementation of generative adversarial networks (GANs) for synthetic data augmentation can further enrich the training corpus, mitigating class imbalance and improving generalization. Furthermore, extending the framework to a multilingual OCR paradigm by incorporating joint embeddings for related Indic scripts will facilitate cross-linguistic model transferability, ensuring broader applicability in low-resource language digitization. This work lays the foundation for a scalable Sanskrit OCR system using novel computational techniques.

5 Acknowledgement

The authors are thankful to PES University, Bengaluru, Karnataka, India.

6 Publisher:

Computer Vision Center
Edifici O - Campus UAB
08193 Bellaterra (Barcelona) - Spain
e-mail: elcvia@cvc.uab.es
Tel: 34 - 93 - 581 18 28
Fax: 34 - 93 - 581 16 70

References

[1] V. A. Raj, R. L. Jyothi and A. Anilkumar, "Grantha script recognition from ancient palm leaves using histogram of orientation shape context," *2017 International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India, 2017, pp. 790-794. https://doi.org/10.1109/iccmc.2017.8282574

[2] A. Tripathi, S. S. Paul and V. K. Pandey, "Standardisation of stroke order for online isolated Devanagari character recognition for iPhone," *2012 IEEE International Conference on Technology Enhanced Education (ICTEE)*, Amritapuri, India, 2012, pp. 1-5. https://doi.org/10.1109/ictee.2012.6208657

[3] N. Sethi, A. Dev and P. Bansal, "A Bilingual Machine Transliteration System for Sanskrit-English Using Rule-Based Approach," *2022 4th International Conference on Artificial Intelligence and Speech Technology (AIST)*, Delhi, India, 2022, pp. 1-5. https://doi.org/10.1109/aist55798.2022.10064993

[4] A. Negi and C. K. Chereddi, "Candidate search and elimination approach for Telugu OCR," TENCON 2003. Conference on Convergent Technologies for Asia-Pacific Region, Bangalore, India, 2003, pp. 745-748 Vol.2. https://doi.org/10.1109/tencon.2003.1273278

[5] L. Rabiner and B. Juang, "An introduction to hidden Markov models," in IEEE ASSP Magazine, vol. 3, no. 1, pp. 4-16, Jan 1986. https://doi.org/10.1109/massp.1986.1165342

[6] R. Chauhan, K. K. Ghanshala and R. C. Joshi, "Convolutional Neural Network (CNN) for Image Detection and Recognition," 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), Jalandhar, India, 2018, pp. 278-282. https://doi.org/10.1109/icsccc.2018.8703316

[7] A. Das, A. S. Azad Rabby, I. Kowsar and F. Rahman, "A Deep Learning-based Unified Solution for Character Recognition," *2022 26th International Conference on Pattern Recognition (ICPR)*, Montreal, QC, Canada, 2022, pp. 1671-1677. https://doi.org/10.1109/icpr56361.2022.9956348

[8] J. Mehta and N. Garg, "Offline Handwritten Sanskrit Simple and Compound Character Recognition Using Neural Network," in Proceedings of International Conference on ICT for Sustainable Development, vol. 408, Springer, Singapore, 2016, pp. 62. https://doi.org/10.1007/978-981-10-0129-1_62

[9] N. Palrecha, A. Rai, A. Kumar, S. Srivastava and V. Tyagi, "Character segmentation for multi lingual Indic and Roman scripts," *2011 IEEE 7th International Colloquium on Signal Processing and its Applications*, Penang, Malaysia, 2011, pp. 45-49. https://doi.org/10.1109/cspa.2011.5759840

[10] M. Sonkusare, R. Gupta, and A. Moghe, "A Review on Character Segmentation Approach for Devanagari Script," in Intelligent Systems, Algorithms for Intelligent Systems, Springer, Singapore, 2021, pp. 19. https://doi.org/10.1007/978-981-16-2248-9_19

[11] S. Kumar, "A study for handwritten Devanagari word recognition," *2016 International Conference on Communication and Signal Processing (ICCSP)*, Melmaruvathur, India, 2016, pp. 1009-1014. https://doi.org/10.1109/iccsp.2016.7754301

[12] A. Dwivedi, R. Saluja and R. K. Sarvadevabhatla, "An OCR for Classical Indic Documents Containing Arbitrarily Long Words," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Seattle, WA, USA, 2020, pp. 2386-2393. https://doi.org/10.1109/cvprw50498.2020.00288

[13] Gadade, Y. A. (2019). "Improving Accuracy of the Tesseract," in limitlessdatascience.wordpress.com. https://limitlessdatascience.wordpress.com/2019/05/01/improving-accuracy-of-the-tesseract/ (Accessed Feb. 23 2024)

[14] R. Shah, M. K. Gupta and A. Kumar, "Ancient Sanskrit Line-level OCR using OpenNMT Architecture," *2021 Sixth International Conference on Image Information Processing (ICIIP)*, Shimla, India, 2021, pp. 347-352. https://doi.org/10.1109/iciip53038.2021.9702666

[15] M. Avadesh and N. Goyal, "Optical Character Recognition for Sanskrit Using Convolution Neural Networks," *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, Vienna, Austria, 2018, pp. 447-452. https://doi.org/10.1109/das.2018.50

[16] C. Halder and K. Roy, "Word & Character Segmentation for Bangla Handwriting Analysis & Recognition," *2011 Third National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics*, Hubli, India, 2011, pp. 243-246. https://doi.org/10.1109/ncvpripg.2011.68

[17] Y. Gurav, P. Bhagat, R. Jadhav and S. Sinha, "Devanagari Handwritten Character Recognition using Convolutional Neural Networks," *2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, Istanbul, Turkey, 2020, pp. 1-6. https://doi.org/10.1109/icecce49384.2020.9179193

[18] S. Kulkarineetham and V. Thananant, "Thai Handwritten Character Recognition Using Deep Convolutional Neural Network," *2023 8th International Conference on Computer and Communication Systems (ICCCS)*, Guangzhou, China, 2023, pp. 1036-1042. https://doi.org/10.1109/icccs57501.2023.10151078

[19] Z. Li, L. Jin and S. Lai, "Rotation-Free Online Handwritten Chinese Character Recognition using Two-Stage Convolutional Neural Network," *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Niagara Falls, NY, USA, 2018, pp. 205-210. https://doi.org/10.1109/icfhr-2018.2018.00044

[20] K. Mokhtar, S. S. Bukhari and A. Dengel, "OCR Error Correction: State-of-the-Art vs an NMT-based Approach," *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, Vienna, Austria, 2018, pp. 429-434. https://doi.org/10.1109/das.2018.63

[21] L. Yu, "State-of-the-art OCR Algorithm for Translation Guiding: An Online-offline Combinational Model," *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*, Coimbatore, India, 2021, pp. 213-217, doi: 10.1109/ICIRCA51532.2021.9544563.

[22] K. C. Santosh, C. Nattee and B. Lamiroy, "Spatial Similarity Based Stroke Number and Order Free Clustering," *2010 12th International Conference on Frontiers in Handwriting Recognition*, Kolkata, India, 2010, pp. 652-657. https://doi.org/10.1109/icfhr.2010.107

[23] "Guest editorial vapnik-chervonenkis (vc) learning theory and its applications," in *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 985-987, Sept. 1999. https://doi.org/10.1109/tnn.1999.788639

[24] S. Singh and M. Sachan, "Opportunities and Challenges of Handwritten Sanskrit Character Recognition System," in *International Journal of Computer Applications*, vol. 3, pp. 18-22, 2017.

[25] C. Hebbi and H. Mamatha, "Comprehensive Dataset Building and Recognition of Isolated Handwritten Kannada Characters Using Machine Learning Models" in *Artificial Intelligence and Applications*, 2023, 1(3), 163-174. https://doi.org/10.47852/bonviewAIA3202624

[26] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," *CVPR 2011*, Colorado Springs, CO, USA, 2011, pp. 1521-1528. https://doi.org/10.1109/cvpr.2011.5995347

[27] G. Dhruva, V. Kore, M. Vijitha, S. Rao, and P. Preethi, "Comprehensive dataset building of isolated handwritten Sanskrit characters," in *Lecture notes in networks and systems*, 2024, pp. 489–503. https://doi.org/10.1007/978-981-97-2004-0_35

[28] Z. Zhao and Q. Ma, "A novel method for image clustering," *2014 10th International Conference on Natural Computation (ICNC)*, Xiamen, China, 2014, pp. 648-652. https://doi.org/10.1109/icnc.2014.6975912

[29] A. Alkandari and S. J. Aljaber, "Principle Component Analysis algorithm (PCA) for image recognition," *2015 Second International Conference on Computing Technology and Information Management (ICCTIM)*, Johor, Malaysia, 2015, pp. 76-80. https://doi.org/10.1109/icctim.2015.7224596

[30] K. P. Sinaga and M. -S. Yang, "Unsupervised K-Means Clustering Algorithm," in *IEEE Access*, vol. 8, pp. 80716-80727, 2020. https://doi.org/10.1109/access.2020.2988796

[31] A. K. Bathla, S. K. Gupta and M. K. Jindal, "Challenges in recognition of Devanagari Scripts due to segmentation of handwritten text," *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, New Delhi, India, 2016, pp. 2711-2715.

[32] N. K. Garg, L. Kaur and M. K. Jindal, "A New Method for Line Segmentation of Handwritten Hindi Text," *2010 Seventh International Conference on Information Technology: New Generations*, Las Vegas, NV, USA, 2010, pp. 392-397. https://doi.org/10.1109/itng.2010.89

[33] M. I. Bhat, B. Sharada, S. M. Obaidullah and M. Imran, "Towards Accurate Identification and Removal of Shirorekha from Off-line Handwritten Devanagari word Documents," *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Dortmund, Germany, 2020, pp. 234-239. https://doi.org/10.1109/icfhr2020.2020.00051

[34] A. B. Shinde and Y. H. Dandawate, "Shirorekha extraction in Character Segmentation for printed devanagri text in Document Image Processing," *2014 Annual IEEE India Conference (INDICON)*, Pune, India, 2014, pp. 1-7. https://doi.org/10.1109/indicon.2014.7030535

[35] R. Ghosh and P. P. Roy, "Comparison of Zone-Features for Online Bengali and Devanagari Word Recognition Using HMM," *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Shenzhen, China, 2016, pp. 435-440. https://doi.org/10.1109/icfhr.2016.0087

[36] M. Vijitha, K. Vrinda, G. Dhruva, R. Sahana, and P. Preethi, "Segmentation of handwritten Sanskrit words using Image-Processing techniques," in *Algorithms for intelligent systems*, 2024, pp. 13–27. https://doi.org/10.1007/978-981-97-5791-6_2

[37] N. Aneja and S. Aneja, "Transfer Learning using CNN for Handwritten Devanagari Character Recognition," *2019 1st International Conference on Advances in Information Technology (ICAIT)*, Chikmagalur, India, 2019, pp. 293-296. https://doi.org/10.1109/icait47043.2019.8987286

[38] S. Patnaik, S. Kumari and S. Das Mahapatra, "Comparison of deep CNN and ResNet for Handwritten Devanagari Character Recognition," *2020 IEEE 1st International Conference for Convergence in Engineering (ICCE)*, Kolkata, India, 2020, pp. 235-238. https://doi.org/10.1109/icce50343.2020.9290637

[39] S. S. Magare and R. R. Deshmukh, "Offline Handwritten Sanskrit Character Recognition Using Hough Transform and Euclidean Distance," International Journal of Innovation and Scientific Research, vol. 10, no. 2, pp. 295-302, Oct. 2014.