



A Comparative Study of Machine Learning and Deep Learning Approaches for Handwritten Sanskrit Recognition

Shraddha V. Shelke¹, Dinesh M. Chandwadkar², Sunita P. Ugale³ and Rupali V. Chothe⁴

ABSTRACT

Sanskrit holds immense historical, cultural, and scientific value, with many handwritten manuscripts preserving knowledge in philosophy and traditional medicine like Ayurveda. It is essential to digitize these materials to preserve this legacy and provide broader access. Handwritten Sanskrit recognition remains an underexplored yet essential area within Optical Character Recognition (OCR), largely due to the complexity of the script. Elements such as the Shirolekha (headline), compound characters, modifiers, and irregular character boundaries contribute to the difficulty of accurate recognition. This paper presents a comprehensive study of classical and contemporary techniques adopted for Sanskrit and Devanagari script recognition. The study explores machine learning methods such as Support Vector Machines (SVM), k-Nearest Neighbors (KNN), and Hidden Markov Models (HMM), alongside modern deep learning models including CNNs, LSTMs, Bidirectional LSTMs, and CapsNet. The analysis shows that while traditional methods work well with limited data, deep learning approaches achieve higher accuracy, like CNN-based architectures, which reach up to 99.65% but require substantially larger datasets and risk overfitting. A key finding is that most existing research focuses on isolated characters, leaving word-level recognition and complex conjuncts largely unaddressed. Persistent issues such as image noise, overlapping characters, and the lack of large-scale Sanskrit word datasets continue to hinder progress. This study highlights existing gaps and proposes future directions, emphasizing the need for hybrid deep learning models, linguistic context via NLP, and benchmark datasets for improved OCR systems.

Article information:

Keywords: Handwritten Sanskrit Recognition, Devanagari Script, Optical Character Recognition, Deep Learning, CNN

Article history:

Received: August 28, 2025

Revised: March 3, 2026

Accepted: May 21, 2026

Published: June 13, 2026

(Online)

DOI: 10.37936/ecti-cit.2026203.263420

1. INTRODUCTION

The Devanagari script is used to write many of India's official languages, including Hindi, Marathi, Sindhi, Nepali, Sanskrit, and Konkani. Devanagari-like scripts are used in several other Indian languages, including Gujarati, Punjabi, and Bengali. Researchers discover a wealth of information regarding science, mathematics, Hindu mythology, Indian history, and culture within ancient Sanskrit manuscripts. Sanskrit holds immense significance as the foundation of many ancient Indian knowledge systems, including medicine (Ayurveda), astronomy, mathematics, and philosophy. Its scientifically structured grammar has influenced modern linguistics and

computational language models. As the language of the Vedas, epics, and classical literature, it embodies India's cultural and intellectual heritage. Globally, Sanskrit continues to attract scholarly interest, yet many of its handwritten manuscripts remain inaccessible. Recognizing and digitizing Sanskrit are essential to preserve this knowledge and making it available for research, education, and cultural preservation. Over 300 million people worldwide use scripts derived from or related to Devanagari, which is also used for Sanskrit [1]. Ancient Sanskrit manuscripts, including those written on palm leaf, are in poor condition, and the National Mission for Manuscripts (NMM) is digitizing this precious literature, but there are still many documents to digitize. It is required to

^{1,2,3,4}The authors are with the K.K Wagh Institute Of Engineering Education and Research, Nashik SPPU Pune, Maharashtra, India, Email: svshelke@kkwagh.edu.in, dmchandwadkar@kkwagh.edu.in, spugale@kkwagh.edu.in and rvchothe@kkwagh.edu.in

¹Corresponding author: svshelke@kkwagh.edu.in

design a better OCR that can recognize not only individual Sanskrit letters but also words and sentences, because the majority of existing OCR systems only work on identifying single Sanskrit characters [2].

Even though the initial research on recognizing the Devanagari script was reported in 1977, research on recognizing handwritten characters and words in Indian scripts has received a lot of interest recently [3]. Numerous OCRs have been developed recently for a variety of Indian languages, including Hindi, Telugu, Gurmukhi, Bangla, etc. However, there has been relatively little work done to create effective OCRs for Sanskrit [4]. Both the machine learning and the deep learning approaches are employed in the design of classifiers. R. Dineshkumar and Dr. J. Suganthi discuss an overview of OCR improvements in several Indian scripts, including Telugu, Hindi, Tamil, Bangla, Gurmukhi, and Devanagari, up to 2013 in [5].

A “Shirorekha” (header line) appears above each character in the Sanskrit script. Sanskrit script consists of 39 Consonants and 16 vowels as shown in Figures 1a and 1b. Vowels occur as modifiers in words and are also called allographs. Conjunct consonants frequently create compound characters by combining several half-letter and full-letter consonants [6], as shown in Figure 1c. Each Indian script has distinct compositional guidelines for how vowels, consonants, and modifiers should be combined. Both vowels and consonants can have a modifier attached to them. In the case of conjunct consonants, a consonant can be represented in half form when combined with another consonant. For some conjunct consonants, the right half part is removed, and the left half part of the first consonant is written as it is [1]. Sometimes the first consonant is represented in the lower half of the conjunct consonant, as shown in Figure 1c.

1.1 Challenges in recognizing Handwritten Sanskrit

a. Structural Complexities in Sanskrit Character Formation

One of the major challenges of recognizing handwritten Sanskrit is the large number of consonants, vowels, and different ways of representing conjunct consonants, as shown in Figure 1. In contrast to English, which has only 26 letters, Sanskrit has a large character set of 55 fundamental characters and is further complicated by over 1,000 possible conjunct consonant combinations. As a result, classifiers must handle thousands of different classes.

b. Script-specific Features

There are three main script-specific features of the Sanskrit script that make the segmentation process challenging. First, the Shirolekha (header line) often connects multiple characters, making individual character isolation difficult. Second, vowel modifiers (allographs) can appear above, below, before, or after consonants, creating spatial relationships that vary

अ	आ	इ	ई	उ	ऊ
		ए	ऐ	ओ	औ
ऋ	ॠ	ऌ	ॡ	अं	अः
(a)					
क	ख	ग	घ	ङ	
च	छ	ज	झ	ञ	
ट	ठ	ड	ढ	ण	
त	थ	द	ध	न	
प	फ	ब	भ	म	
	य	र	ल	व	
	श	ष	स	ह	
	ळ	क्ष	त्र	ज्ञ	
	श्री	ॐ			
(b)					
क्ख	क्का	क्कघ			
ग्य	ग्र	गल			
त्य	तल	तळ			
प्य	प्र	प्ल			
(c)					

Fig.1: (a) Vowels in the Sanskrit script, (b) Consonants in the Sanskrit script and (c) Compound characters / Conjunct consonants.

across different character combinations. Third, conjunct consonants frequently exhibit non-linear boundary formations where characters partially overlap or merge, as illustrated in Figure 2(b).

c. Comparative Analysis with Other Languages

Unlike scripts such as Latin or Korean, where character boundaries are relatively well-defined, Sanskrit’s complex orthography produces segmentation ambiguities rarely encountered in other writing systems [7]. While Arabic script presents its own challenges with cursive connections, Sanskrit’s combination of horizontal connections (through Shirorekha) and vertical stacking (in conjuncts) creates multi-dimensional segmentation problems. Similarly, although Chinese characters offer visual complexity, they typically maintain consistent spatial boundaries, whereas Sanskrit characters deform and reorganize when forming conjuncts.

d. Handwriting Variability and Segmentation Challenges

Different people have different handwriting styles, which can change over time. Slant, a particular shape, average letter size, thickness, pressure applied

to the paper, and regular or atypical spacing between characters are the features of handwriting. These features help to identify the handwritten characters and contribute to writing discrepancies, as shown in Figure 2(c) [8].

Challenges in handwritten Sanskrit recognition include Shirolekha detection, detection of top, middle, and bottom zones shown in figure 2 (a), Modifiers in these zones, Non-linear boundaries between two adjoining characters, and writing discrepancies. Finding the boundary between characters that are touching each other and creating a single cluster is difficult to estimate.

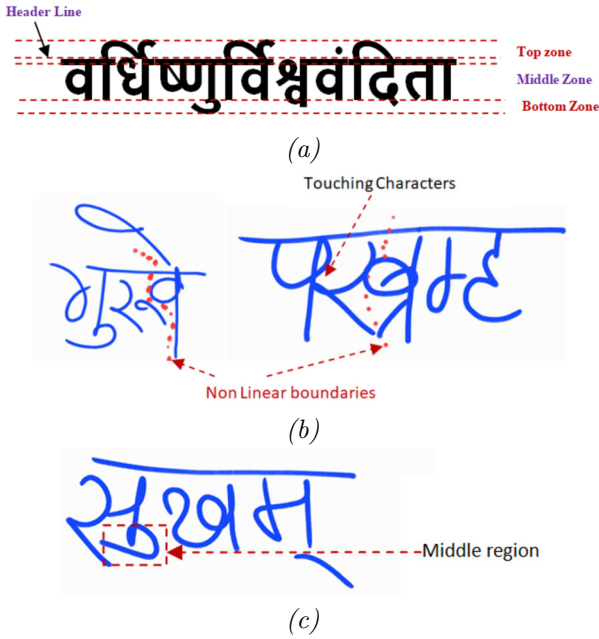


Fig.2: (a) Top, middle, and bottom zones of a Sanskrit character, (b) Non-Linear boundaries and (c) Writing discrepancies.

Having a good quality input image is also one of the important parameters to reduce recognition error. Errors can be due to the aging of palm leaves, complex backgrounds, skewness, change in text layout / fonts, including special characters, blurring, and degradation, etc [9]. Therefore, a system that addresses these issues and delivers greater accuracy while requiring shorter recognition time must be developed [4], [9].

While OCR has advanced significantly for English and other languages, Sanskrit hasn't received the same attention. Most studies so far focus on isolated characters or small datasets, and even when they look at the Devanagari script, they tend to avoid the complex conjuncts that make Sanskrit so challenging. There is a need for comparison of traditional machine learning methods like SVM and HMM side by side with modern deep learning approaches like CNNs and RNNs, and seeing how they perform on Sanskrit's unique challenges. This paper tries to fill

that gap by comparing these methods under the same conditions, so we can finally have some clear benchmarks and help researchers choose the right approach for their specific needs.

2. CHARACTER RECOGNITION TECHNIQUES

Techniques broadly used for recognizing handwritten Devanagari and Sanskrit scripts are categorized into segmentation-based and segmentation-free approaches.

2.1 Segmentation-based approach

Alternatively referred to as an analytical approach, the segmentation-based approach breaks down each word into discrete equivalent components (such as character parts or even character subparts). It then identifies each equivalent component (or character), gives it a unique symbol, and reconstructs the symbols to determine the word [9]. The segmentation-based technique employs several algorithms, including projection-based methods [10], Connected Component Analysis [11], [12], Clustering Methods (KNN) [13], Neural Networks [14], Fuzzy Logic [15], and Hidden Markov Models (HMMs) [16], [17].

Bikash Shaw *et al.* proposed a unique segmentation-based technique for recognizing offline handwritten Devanagari words [18]. The work of segmentation is fairly challenging due to the wide range of handwriting styles and the cursiveness of the script. For the segmentation process, they assumed that a header line (Matra) is present in every word. Segmentation is easier for printed characters than for handwritten ones because header lines in handwritten script are not necessarily straight or parallel to each other. Bikash Shaw and *et al.* used a string edit distance technique for word-level recognition. Using a segmentation-based technique, R. Jayadevan *et al.* suggested a system to identify handwritten legal amounts found on Indian bank checks written in Hindi and Marathi [10]. In this study, the method divides words into connected components and measures the distance between two successive connected components. This study uses the vertical projection profile (VPP) approach to segment characters from words.

Another method for segmenting and identifying touching machine-printed Devanagari characters was provided by Garain and Chaudhuri [15]. Fuzzy multifactorial analysis is used in this approach. For segmenting touching characters, the system applies a predictive algorithm. Namita Dwivedi *et al.* proposed a genetic algorithm for segmentation in [19]. This study also defines the individual coding, fitness function, crossover operator, and mutation operator for the segmentation task.

2.2 Segmentation Free approach

In a segmentation-based approach, there is no need for the segmentation of words into individual characters. This approach is also called a holistic approach, where the system processes every word as a whole unit. In this approach, after extracting features of the whole set of words, the training classifier is used for classification. This method has problems with complexity and a lack of discrimination capacities because it processes the entire word as one indivisible unit [9]. Segmentation-free approach includes Recurrent Neural Networks [20], Long Short-Term Memory Networks [21], Convolutional Neural Networks [2], [22], [23].

S. K. Parui and B. Shaw proposed a robust approach that is independent of many input factors like thickness, size, etc. During the pre-processing smoothing phase, a median filter first removes noise from the input grey-level image, and then Otsu's thresholding approach binarizes the image. The system then extracts horizontal and vertical stroke information from word images and classifies them using Hidden Markov Models (HMM). The authors found promising results for a small Lexicon size [24]. However, the authors do not work on skew and slant correction.

A segmentation-free approach for recognizing printed Devanagari script using Long Short-Term Memory (LSTM) networks is proposed by Tushar Karayila *et al.* [21].

The results of using LSTM networks on the Devanagari script, where each consonant-consonant conjunct and consonant-vowel combination has a different form depending on where it occurs in the word, are reported in this research. The LSTM network underwent two phases of training. The first one was using only images from a single font, and the second

one used a training set that included images from other fonts. When the test fonts matched those in the training data, an LSTM-based system produced a character error rate of 1.2%, but when tested on scanned data, the error rate increased to (9%). To improve results, 2DLSTM can be used as these nets would scan the text lines not only sideways, but also top-down.

B. Shaw *et al.* proposed another approach to recognizing handwritten Devanagari words using continuous density Hidden Markov Models [16]. In this work, the feature vector is obtained by computing a chain-code directions histogram from a sliding window that moves across the input image from left to right. A database of handwritten city names is used to determine the states of the HMMs automatically.

3. RELATED WORK

Handwritten or printed characters/symbols, words are recognized and classified by either the template matching method or the feature extraction method. By comparing an unknown test pattern to the ideal pattern, the template-based technique allows for classification and decision-making based on the degree of correlation between the two patterns. In the case of feature extraction, various features are extracted from input images, such as statistical, structural, Histogram of Oriented Gradients (HOG), [25], [26], Chain Code Histogram [27], Wavelet [28], [29]. Zernike moment feature descriptor [30], [31] etc. are used. Classifiers like Neural Networks (NN) [11], support vector machines (SVM) [31], Convolutional Neural Networks [2], [22], [23], and hidden Markov models (HMM) [16], [17], [32] are used for classification tasks. The recognition of Devanagari characters also uses certain modern methods, such as fuzzy rules [33], evolutionary algorithms, etc. [1]. Beyond neural net-

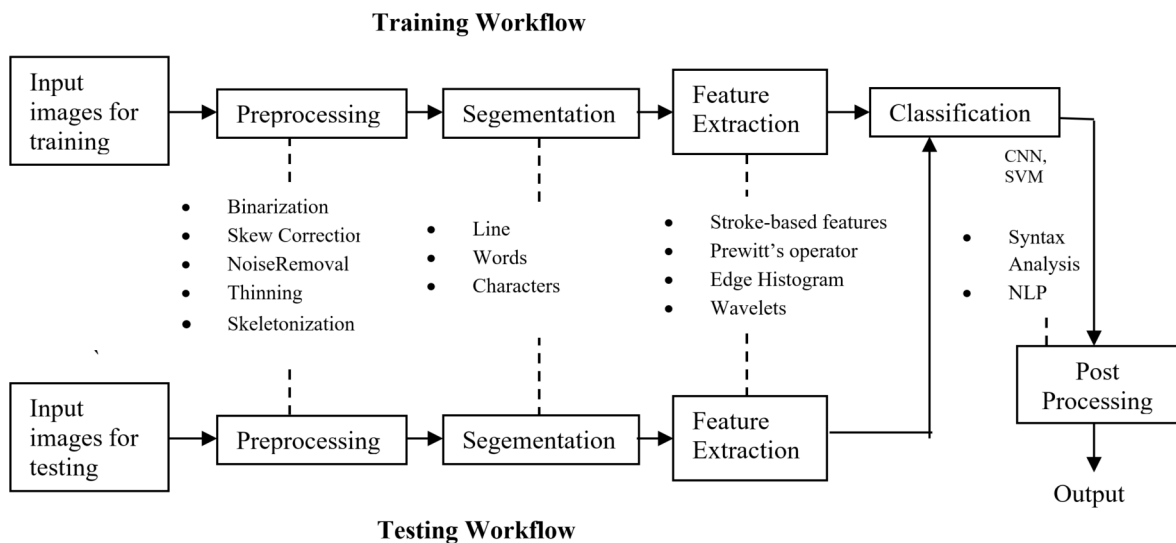


Fig.3: General Workflow of Handwritten Script Recognition.

works, traditional ML classifiers — including k-NN [13], SVM [19], and Random Forest — have been extensively explored for Devanagari recognition. These classifiers often offer complementary strengths, such as handling limited training data and providing interpretable decision boundaries.

Recognizing handwritten or printed scripts involves three primary steps. First, Document layout analysis, region extraction, and categorization. Second, Segmentation of text lines, words, and characters -finally, Recognition of individual character. Figure 3 shows the general workflow of handwritten script recognition.

3.1 Handwritten Character Datasets available

Devanagari Handwritten Character Dataset (DHCD) collected by Acharya *et al.* [34] Includes 92,000 images of handwritten Devanagari character. It includes 72,000 images of consonants and vowels and 20,000 images of numerals from 0 to 9. There are 2000 samples of every character image from 46 classes. ($2000 \times 46 = 92,000$). Figure 4(a) shows sample images from the DHCD dataset. This dataset comprises images that showcase a broad spectrum of writing styles, cursive handwriting, and variations in thickness. Researchers can access the NHCD dataset, compiled by A. K. Pant *et al.* [35]. It contains handwritten Devanagari character samples that are taken from 40 people in various fields and cropped for character boundaries. The dataset includes 12,912 images of numerals, consonants, and vowels. Sample images are shown in Figure 4(b). Another dataset, collected by S. Deore and Albert Pravin [2], contains RGB images of handwritten Devanagari consonants, vowels, and numerals across 58 classes and is available for research. The dataset includes 5800 handwritten images with different writing styles. Sample images are shown in Figure 4(c).

Notably, reported accuracy varies significantly based on experimental conditions. Character-level tasks [2], [22] typically achieve higher accuracy (94-99%) than word-level recognition [11], [36] (70-97%). Another important factor is the size of the dataset; models trained on larger datasets, such as DHCD (92,000 images) [34], typically have higher generalization than models using smaller collections [23].

3.2 Review of Preprocessing and Skew Correction

During the preprocessing and segmentation phase, a small degree of skew is present when a document is scanned. One of the important steps in layout analysis is the estimation and correction of the skew (tilt) angle. Detection of the header line (Shirorekha) can be used for skew estimation in the Sanskrit script. B. Chaudhuri and U. Pal [12] proposed a skew angle detection approach by considering that the skew of the

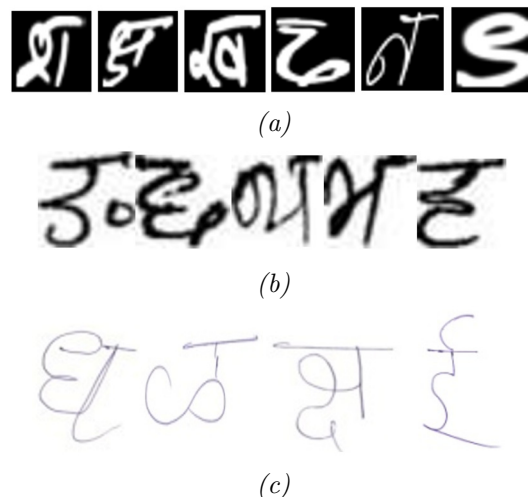


Fig.4: (a) Sample images of the DHCD dataset, (b) NHCD Dataset and (c) S. Deore dataset.

shirolekha shows the skew of the entire document. They define a bounding box for each connected component after labeling it. The text also includes dots, isolated characters, punctuation marks, tables, etc. There is a need to remove this information as it may introduce errors while skewing detection. The mean and standard deviation of the bounding box are calculated, and the filtering process is performed based on them to remove such information. To determine the skew, straight line segments are located in the upper envelopes of bounding boxes. This method averages the angles between the horizontal line and the lines joining the first pixel of the leftmost line to obtain the skew angle. In the case of multiscale documents, text lines are not parallel to each other.

U. Pal, M. Mitra, and B. Chaudhuri [37] describe a robust algorithm extension to previous algorithms for multiscale angle detection in Devanagari and Bangla scripts with an accuracy of 98.3%. A column-wise scan starting at the top of the component determines the upper envelopes of selected components. After performing skew estimation, the system separates text lines, words, and characters for effective feature extraction. After extracting text blocks from scanned documents, line segmentation and word segmentation are done [1].

3.3 Comparative Analysis of Classification Models

A two-stage classification approach tested on 50000 samples with 89.21% accuracy is proposed by S. Arora *et al.* [14]. This technique uses a differential distance-based method to detect the Shirorekha and the vertical bar (spine). A feed-forward neural network is used for classification purposes. The proposed algorithm tested characters by dividing them into three groups based on Shirorekha continuity: end spine, mid spine, and no spine. An edge detec-

tion algorithm based on Prewitt's operation to recognize handwritten Sanskrit words is proposed by N. Dwivedi *et al.* [19]. The thinning algorithm is used to produce the skeleton of the character in the pre-processing stage. This method uses the Freeman chain code (FCC) as a feature extraction technique. During FCC generation, the problem of finding several branches and revisiting the same nodes, again and again, is solved by generating heuristics. Figure 5 shows the proposed system architecture. A genetic algorithm includes SVM as a classifier that works for classifying single and multiple characters.

Satish Kumar [11] implemented a segmentation-based approach using the multi-layer perceptron (MLP) as a classifier. Algorithm is applied on 3500 words which are written by 200 different people in Devanagari script. Two cases considered during segmentation are straight shirolekha words and skewed shirolekha words. Non linear boundaries between characters shown in figure 2(a) is one of the major challenge in segmentation is addressed in this algorithm.

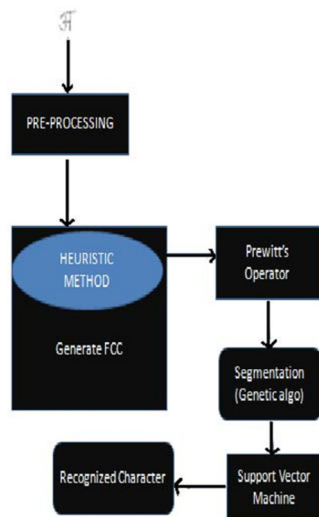


Fig.5: Sanskrit Character recognition system architecture implemented in [19] using Prewitt's operator and SVM.

A connected component technique based on chain code is used to extract every character in the middle part of the word. Researchers apply the width estimation and recognition technique to handle touching characters. The system moves a box—whose size matches the tentative size of a character—over the merged letters. Only correctly segmented words are used for the classification process. Classification results are obtained for a word with different character sizes from 2 to 6 characters. Overall segmentation results for two-character words are 80.8% and for six-character words, it is 70%. However, the authors found scope to improve the segmentation rate for characters that touch each other, join together, or have tilted headlines.

To improve the overall effectiveness of the Devanagari handwritten character recognition system (DHCRS), a two-stage deep learning approach was developed by S. Deore and Albert Pravin. The system tested on 5800 isolated images of 58 unique characters as a dataset. The testing accuracy of the first model is 94.84% with a loss of 0.18. However, the second model, which requires very little training time, correctly classified 96.55% of instances with a training loss of 0.12. Additionally, authors obtained encouraging outcomes from the suggested model when it came to testing four distinct benchmark datasets that had isolated characters and Indic script digits. The proposed model, according to the authors, had extensive memory because it had more training parameters [2].

After comparing these approaches, several key observations are taken. CNN-based methods [2], [22], and [23] consistently achieve better accuracy (94–99%), but they need a lot of training data and processing power, and they run the risk of overfitting when datasets are small [22]. Conventional machine learning techniques, such as SVM [19] and HMM [16], are more interpretable and work well with smaller datasets, but they have trouble handling the high variability of handwritten Sanskrit. There are trade-offs when choosing between segmentation-based [11], [18] and segmentation-free [21], [24] approaches: segmentation-based methods offer modularity but compound errors, whereas holistic approaches maintain contextual information but encounter difficulties with big words.

P. Gupta *et al.* [22] proposed a Convolutional Neural Network approach using a dataset of 378,951 characters and applied data augmentation to generate images. This approach implemented online-based models with six layers—including two convolutional layers, two max-pooling layers, and two fully connected layers—achieving an accuracy of 95.6%. The architecture used kernel sizes of 3×3 and 5×5 in convolutional layers, with ReLU activation and a dropout rate of 0.25 to mitigate overfitting despite the large training dataset. The drawback of this approach is that there may be instances of overfitting because of the large size of the training dataset.

Y. Gurav *et al.* [23] implemented another offline-based model using Deep Convolutional Neural Networks (DCNN). The dataset includes 300 standard binarized images of size 28×28 . The system trained images for 15 iterations. The authors achieved an accuracy of 99.65%. This approach uses consecutive convolutional layers, which has an added benefit for the process of obtaining higher-level features. The DCNN architecture employed 3×3 convolutional kernels, 2×2 max pooling, and a dropout of 0.5, and was trained using stochastic gradient descent with a learning rate of 0.001. The authors introduced a dropout layer to prevent overfitting. From the confusion matrix, the close resemblance between a few characters

increases the chances of wrong prediction.

Lomte and Doye implemented three different CNN models with different convolutional layers, as well as the AlexNet model, to recognize Sanskrit words from a dataset of 7000 words, achieving an accuracy of 97.42% and an average recognition time of 0.364 ms [36]. The authors found better results than others af-

ter testing their algorithm on the same dataset. However, the method gives less accuracy for blurred and overlapped text images. Figure 6 shows an 8-layer CNN that uses 3×3 filters throughout, with max-pooling after every two convolutional layers and a final softmax layer of 1,000 neurons corresponding to word classes. However, the method gives lower accu-

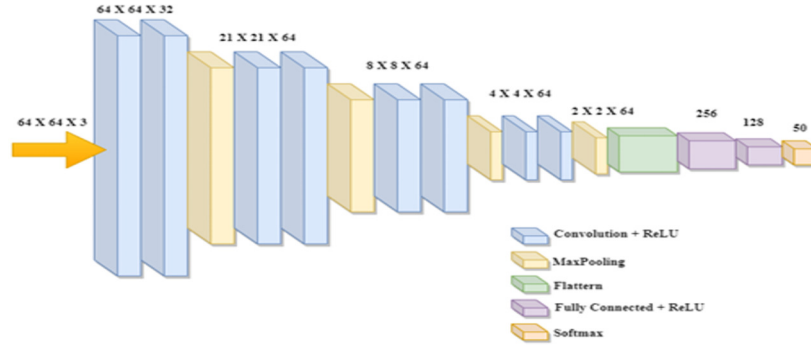


Fig.6: CNN model with 8 convolutional layers proposed in [36] for classification of handwritten Sanskrit Words.

Table 1: The comprehensive study of existing algorithms for handwritten characters and word recognition.

Ref. No.	Year	Characters / Words	Dataset	Methodology used	Classification Accuracy
[17]	2010	Characters	42 stroke classes	Stroke-based approach, HMM	The proposed iPhone app includes three modules: stroke classification using HMMs, stroke identification with lookup tables, and a user interface displaying potential characters.
[10]	2010	Marathi and Hindi Words	26,720	Gradient, structural and cavity (GSC) features with a binary vector matching (BVM) technique. Vertical projection profile (VPP) feature and dynamic time warping (DTW)	85.78%
[32]	2011	Numerals	500	Moment normalization, HMM	N=3 accuracy=30% to 100% N=5, accuracy =80% to 100%
[19]	2013	Words	Unspecified	Prewitt's edge detection operator, Freeman Chain Code, SVM	To accurately produce the FCC and address the issue of repeatedly visiting the same node during the FCC development process, heuristics are employed SVM classifier gives higher classification accuracy.
[38]	2012	Characters	Training: 468 Test: 36	Zoning, MLP, K-Nearest Neighbor, Naive Bayes, Decision Tree	MLP: 77.78% KNN: 63.89% Naive Bayes: 69.44% Decision Tree: 47.22%
[30]	2012	Words	Training: 1190 Testing: 700	Gradient and Zernike moment, Support Vector Machines	False Rejection Rate: 7.42% False Acceptance Rate: 4.28%
[39]	2012	Characters	2760	Discrete Cosine Transform, Neural Network	92.7 %

[31]	2014	Compound Characters	27000	Zernike moment feature descriptor, SVM and KNN	SVM: 98.37% k-NN: 95.82%
[7]	2014	Words	Devnagri: 15,485 Roman: 5,427	Convex hull-based features and Naïve Bayes, Bayes net, support vector machine (SVM), decision table, radial basis function (RBF), MLP, multiclass classifier and ordinal class classifier	Naïve Bayes: 87.72%, Bayes net: 92.37%, SVM: 98.41, Decision table: 98.45%, RBF: 98.33%, MLP: 99.87%, Multiclass classifier: 99.32% Ordinal Class Classifier: 97.14%
[28]	2014	Characters	468	Wavelet families: Daubechies and Reverse Biorthogonal Decision Tree Multilayer perception, K-Nearest Neighbor, Naïve Bayes	Decision Tree: 89%, Naive Bayes: 97%, KNN: 100%, MLP: 98%
[21]	2015	Script	Training: 24000 Test 22517	LSTM	Error rate = 1.2% for test fonts same as training data. The error rate increased by 9% when tested on scanned data
[11]	2016	Words	2 to 6 letter handwritten chart: 3500	Connected component technique based on chain code, MLP	2 char words: 80.8% 3 char words: 80.3% 4 char words: 80.1% 5 char words: 75.3% 6 char words: 70.0%
[40]	2016	Characters	3600	Multilayer perceptron, K-Nearest Neighbor, Naïve Bays, Decision Tree	MLP: 98% KNN:100%, Naïve Bays: 63.3%, Tree J-48: 80%
[41]	2017	Characters	800	Template matching	Consonants: 91.4% Vowels :76.47% Numerals:100%
[26]	2017	Characters	600	1. Histogram of Orientation Shape Context (HOOSC) 2. Histogram of Oriented Gradients (HOG) 3. Gabor Features 4. Zoning 5. Invariant Moments ANN	1. HOOSC: 96.5% 2. HOG: 84% 3. Gabor Features: 73.6% 4. Zoning: 76% 5. Invariant Moments: 52%
[13]	2017	Numerals	300	HOG, PCA, KNN, SVM, ANN	HOG-KNN: 95.27 HOG- SVM: 75.02 HOG-ANN: 95.20 PCA-SVM: 51.02 PCA-KNN: 61.73 PCA-ANN: 57.30
[4]	2018	Simple and compound characters	Training: 10,106 Test: 1124	RGB to Grayscale, Convnet	93.32%
[42]	2018	Words	624	Curvelet Transform, MLP	98.92%
[33]	2018	Compound Characters	Training: Characters 9700, Numerals 2000, Test: 1650	A fuzzy logic-based system using a rotation invariant rule-based thinning algorithm	With thinning: 88 - 95% Without thinning: 85-91%

[29]	2019	Character	12,912	Discrete wavelet transform (DWT) and Moment features SVM, CNN	89.96%
[2]	2020	Character	5,800	Deep Convolutional Neural Network	96.55%
[22]	2020	Characters	3,78,951	Convolutional Neural Network	95.60%
[23]	2020	Characters	34,604	Convolutional Neural Network	99.65%
[43]	2020	Script	200	Global threshold and Otsu's threshold	Segmentation accuracy: 85.12%
[44]	2021	compound characters	5,000	Edge Histogram, SVM, SMO, MLP, and Simple Logistic	SVM: 99.88% SMO: 99.72% Simple Logistic model: 99.04% MLP: 97.7%
[36]	2022	Words	70,000	4-Fold Deep CNN Model	CNN Model: 98.8% Alex Net: 53.75
[45]	2023	Numerals MNIST	Unspecified	Resnet 101V2, Vgg19, Vgg16 and combined model with Adam and SGD optimizer	Combined model: 99.72%
[6]	2023	Words, conjunctions characters and modifiers	9,854	CapsNet model	94.6% for 30 epochs
[46]	2023	Four datasets for Malayalam characters	91,902	LeNet5, CNN1(4 layers with ReLU), CNN2 (16 layers with ReLU)	Maximum accuracy of 99.96%
[25]	2023	Characters	72,000	Resizing of Image, Canny Edge Detection, Hybrid DWT-DCT Feature extraction, Histogram of Oriented Gradients (HOG)Neural Network, Support vector machines (SVM)	HOG-NN: 91.40% HOG-SVM: 97.10%
[20]	2024	Characters	Training: 834,036 Validation: 22,542	CNN-RNN model with Bidirectional Gated Recurrent Units (Bi-GRU)	96.72%
[47]	2025	Characters	72,000	DWT-CNN GoogleNet	98.97% 99.68%

racy for blurred and overlapped text images. Results also shows that adam optimizers gives better classification results than sgdm optimizers. Table 1 provides in-depth research on current methods for handwritten character and word recognition.

4. OBSERVATIONS

he presence of a benchmark dataset is very much essential for successful research. Some datasets are available for Devanagari characters and words, but they do not include complex words like Sanskrit shlokas. Computerized Sanskrit word data sets are available but the handwritten Sanskrit dataset is not available for research [1]. There has been extensive research work carried out on the recognition of handwritten Hindi, Gujarati, Punjabi, Bengali, and Kannada characters. Still, there is scope to develop novel recognition techniques for Sanskrit script due to its

complex structure like presence of Shirolekha, compound characters, modifiers, half-letter characters, similar overlapping and touching characters etc. [9]. Challenges in recognizing ancient Sanskrit grantha script include writing on palm leaves, aging of palm leaves, faint ink strokes, background images, and unwanted impurities [26].

In addition, one of the crucial phases in skew identification is shirolekha detection. The absence of Shirorekha and the straightness of Shirorekha are two concerning issues. Horizontal and vertical line strokes in printed Sanskrit characters are straight, which makes them easier to locate than those in handwritten characters, where such strokes are not always straight. During segmentation-based recognition, lines are separated into words, and then words are further separated into individual characters and modifiers. Recognition accuracy depends on the cor-

rect segmentation process. Proper identification of touching characters and non-linear boundaries improves recognition accuracy. Using contextual information may also improve results.

4.1 Research Gaps Identified from Literature

The studies in Table 1 show that various approaches achieve good accuracy, but they also have limitations. Table 2 highlights these gaps across major works.

The observation shows that high character accuracy often does not lead to good word recognition. Table 2 shows three recurring challenges in recognition, indicating substantial room for further research, which are segmentation difficulties, dataset limitations, and architecture mismatches with Sanskrit's unique structure.

Figure 7 provides a visual comparison of classification accuracies achieved by major models discussed in the literature. Over the past few years, numerous researchers have begun implementing OCR using machine learning and deep learning techniques. Some of

the research shows that convolution neural network-based classifiers are

5. CONCLUSION

Sanskrit manuscript digitization requires advanced OCR methods designed to handle the linguistic complexities and structural difficulties of the Devanagari script. This study highlights that while segmentation-based methods contributed to early progress, they often struggle with non-linear character boundaries and complex compound structures. In contrast, deep learning approaches—particularly convolutional neural networks (CNNs) and their hybrids with recurrent models such as RNNs and LSTMs—demonstrate superior performance in holistic and word-level recognition tasks.

However, recognition accuracy remains highly dependent on the availability of well-annotated datasets and effective preprocessing steps, such as skew correction and zone separation. A major gap persists: limited large-scale handwritten Sanskrit datasets at the word and sentence levels, since most existing re-

Table 2: Research Gaps Identified from Literature.

Ref.	Methodology used	Technical Gap
[2]	Two-stage deep CNN	The model requires significant memory, making deployment on everyday devices difficult.
[6]	CapsNet	CapsNet takes 3-5 times longer to train than standard CNNs.
[7]	Convex hull + 8 classifiers	Two different character shapes can sometimes produce identical convex hulls, causing confusion.
[11]	MLP with connected components	Connected component analysis fails when characters touch vertically rather than just horizontally. Tilted headlines cause segmentation errors.
[19]	Prewitt, FCC, Genetic Algorithm, SVM	The chain code generation process sometimes revisits the same nodes, and there's no guarantee the heuristic will find the optimal path.
[21]	LSTM	LSTM misses 2D spatial structure.
[22]	CNN with $3 \times 3/5 \times 5$ kernels	Fixed kernels miss varying stroke thickness. No data augmentation was applied to handle rotated or scaled characters
[23]	DCNN with 4 layers	Similar-looking characters like ष and ढ are often misclassified.

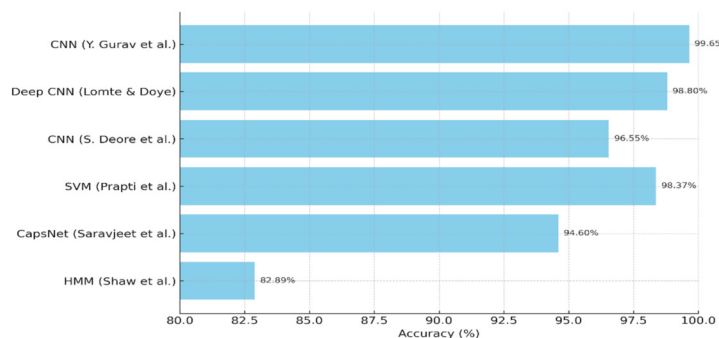


Fig.7: Comparison of classification accuracy across models.

sources focus on isolated characters. Additionally, deep learning models often face challenges like overfitting and reduced robustness when exposed to noisy, blurred, or degraded manuscript images.

To address these challenges, future research must prioritize the development of large-scale, diverse datasets that include shlokas and complex conjunct characters. Integrating image recognition with natural language processing (NLP) can enhance the semantic understanding of handwritten Sanskrit texts. Techniques like 2D-LSTMs and CNN-RNN models, especially when supported by smart data augmentation, can improve accuracy while maintaining computational efficiency. Ultimately, a reliable and scalable OCR system for Sanskrit will be instrumental in preserving ancient knowledge and making India's cultural and intellectual heritage accessible to a global audience.

AUTHOR CONTRIBUTIONS

Conceptualization, S.V.S. and D.M.C.; methodology, S.V.S.; software, S.V.S.; validation, S.V.S., D.M.C. and S.P.U.; formal analysis, S.V.S.; investigation, S.V.S.; data curation, S.V.S.; writing—original draft preparation, S.V.S.; writing—review and editing, S.V.S., D.M.C. and S.P.U.; visualization, S.V.S. and R.V.C.; supervision, D.M.C. All authors have read and agreed to the published version of the manuscript.

References

- [1] R. Jayadevan, S. R. Kolhe, P. M. Patil and U. Pal, "Offline Recognition of Devanagari Script: A Survey," in *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 6, pp. 782-796, Nov. 2011.
- [2] S. P. Deore and A. Pravin, "Devanagari Handwritten Character Recognition using fine-tuned Deep Convolutional Neural Network on trivial dataset," *Sādhana*, vol. 45, p. 243, 2020.
- [3] S. I. K. and B. Chatterjee, "Machine recognition of constrained hand printed Devanagari," *Pattern Recognition*, vol. 9, no. 2, pp. 69-75, 1977.
- [4] 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, pp. 447-452, 2018.
- [5] R. Dineshkumar and J. Suganthi, "A Research Survey on Sanskrit Offline Handwritten Character Recognition," *International Journal of Scientific and Research Publications*, vol. 3, no. 1, 2013.
- [6] A. Moudgil, S. Singh, V. Gautam, S. Rani and S. H. Shah, "Handwritten devanagari manuscript characters recognition using capsnet," *International Journal of Cognitive Computing in Engineering*, vol. 4, pp. 47-54, 2023.
- [7] P. K. Singh, R. Sarkar, N. Das, S. Basu and M. Nasipuri, "Statistical comparison of classifiers for script identification from multi-script handwritten documents," *International Journal of Applied Pattern Recognition*, vol. 1, no. 2, pp. 152-172, 2014.
- [8] S. S. Kumar and S. M. Kumar, "Opportunities and Challenges of Handwritten Sanskrit Character Recognition System," *International Journal on Future Revolution in Computer Science & Communication Engineering*, vol. 3, no.7, pp. 18-22, 2017.
- [9] S. Singh and N. K. Garg, "Review of Optical Devanagari Character Recognition Techniques," in *Intelligent System Design*, Springer, pp. 97-106, 2020.
- [10] R. Jayadevan, S. R. Kolhe, P. M. Patil and U. Pal, "Database Development and Recognition of Handwritten Devanagari Legal Amount Words," *2011 International Conference on Document Analysis and Recognition*, Beijing, China, pp. 304-308, 2011.
- [11] S. Kumar, "A study for handwritten Devanagari word recognition," *2016 International Conference on Communication and Signal Processing (ICCSPP)*, Melmaruvathur, India, pp. 1009-1014, 2016.
- [12] B. B. Chaudhuri and U. Pal, "Skew angle detection of digitized Indian script documents," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 182-186, Feb. 1997.
- [13] A. Chaudhary and R. L. Choker, "Handwritten Hindi numeric character recognition and comparison of algorithms," *2017 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence*, Noida, India, pp. 13-16, 2017.
- [14] S. Arora, D. Bhattacharjee, M. Nasipuri and L. Malik, "A Two Stage Classification Approach for Handwritten Devnagari Characters," *International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)*, Sivakasi, India, pp. 399-403, 2007.
- [15] U. Garain and B. B. Chaudhuri, "Segmentation of touching characters in printed Devnagari and Bangla scripts using fuzzy multifactorial analysis," *Proceedings of Sixth International Conference on Document Analysis and Recognition*, Seattle, WA, USA, pp. 805-809, 2001.
- [16] B. Shaw, S. K. Parui and M. Shridhar, "Offline Handwritten Devanagari Word Recognition: A Holistic Approach Based on Directional Chain Code Feature and HMM," *2008 International Conference on Information Technology*, Bhubaneswar, India, pp. 203-208, 2008.
- [17] S. B. Patil, G. R. Sinha and V. S. Patil, "Isolated Handwritten Devnagri Numeral Recognition Us-

- ing HMM,” *2011 Second International Conference on Emerging Applications of Information Technology*, Kolkata, India, pp. 185-189, 2011.
- [18] B. Shaw, S. K. Parui and M. Shridhar, “Offline handwritten Devanagari word recognition: A segmentation based approach,” *2008 19th International Conference on Pattern Recognition*, Tampa, FL, USA, pp. 1-4, 2008.
- [19] N. Dwivedi, K. Srivastava and N. Arya, “Sanskrit word recognition using Prewitt’s operator and support vector classification,” *2013 IEEE International Conference ON Emerging Trends in Computing, Communication and Nanotechnology (ICECCN)*, Tirunelveli, India, pp. 265-269, 2013.
- [20] S. S. Nair and P. R. J. Thangaiyah, “A Novel Approach: CNN-RNN and Bi-GRU for Handwritten Character Recognition,” *SSRG International Journal of Electronics and Communication Engineering*, vol. 11, no. 1, Jan. 2024.
- [21] T. Karayil, A. Ul-Hasan and T. M. Breuel, “A segmentation-free approach for printed Devanagari script recognition,” in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pp. 946-950, 2015.
- [22] P. Gupta, S. Deshmukh, S. Pandey, K. Tonge, V. Urkunde and S. Kide, “Convolutional Neural Network based Handwritten Devanagari Character Recognition,” *2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)*, Bengaluru, India, pp. 322-326, 2020.
- [23] 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, pp. 1-6, 2020.
- [24] S. K. Parui and B. Shaw, “Offline handwritten devanagari word recognition: An HMM based approach,” in *Proceedings of International Conference on Pattern Recognition and Machine Intelligence (PReMI)*, Springer, 2007.
- [25] S. V. Shelke, D. M. Chandwadkar, S. P. Ugale and R. V. Chothe, “Combining Multiple Feature Extraction and Classification Methods to Study Performance of Handwritten Sanskrit Character Recognition,” *2023 7th International Conference On Computing, Communication, Control And Automation (ICCUBEA)*, Pune, India, pp. 1-6, 2023.
- [26] 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 (IC-CMC)*, Erode, India, pp. 790-794, 2017.
- [27] S. Arora, D. Bhattacharjee, M. Nasipuri and D. K. Basu, “Recognition of Non-Compound Handwritten Devnagari Characters using a Combination of MLP and Minimum Edit,” *International Journal of Computer Science and Security*, vol. 4, no. 1, pp. 1-14, 2010.
- [28] S. V. Shelke, A. H. Dhangare and S. D. Patil, “Handwritten Character Recognition using Wavelet Transform for Feature Extraction,” *International Journal of Multidisciplinary Educational Research*, vol. 3, no. 7, pp. 118-121, 2014.
- [29] J. P. S. and N. G. Chitaliya, “Machine Learning and Deep Learning Approaches for Sanskrit Character Recognition,” *Journal of Xi’an Shiyou University*, vol. 15, no. 3, pp. 48-54, 2019.
- [30] S. Pal, M. Blumenstein and U. Pal, “Hindi Off-Line Signature Verification,” *2012 International Conference on Frontiers in Handwriting Recognition*, Bari, Italy, pp. 373-378, 2012.
- [31] K. V. Kale, P. D. Deshmukh, M. M. Kazi and Y. S. Rode, “Zernike Moment Feature Extraction for Handwritten Devanagari (Marathi) Compound Character Recognition,” *International Journal of Advanced Research in Artificial Intelligence*, vol. 3, no. 1, pp. 68-76, 2014.
- [32] S. Patil, G. R. Sinha and V. S. Patil, “Isolated handwritten Devnagri Numeral recognition using HMM,” in *Proceedings of International Conference on Emerging Applications of Information Technology*, pp. 185-189, 2011.
- [33] P. Yawalkar and M. U. Kharat, “Recognition of Hand Written Devnagri Characters Using Effective Thinning and Fuzzy Logic,” *Journal of Advanced Research in Dynamical & Control Systems*, vol. 10, no. 15, 2018.
- [34] S. Acharya, A. K. Pant and P. K. Gyawali, “Deep learning based large scale handwritten Devanagari character recognition,” *2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, Kathmandu, Nepal, pp. 1-6, 2015.
- [35] A. K. Pant, S. P. Panday and S. R. Joshi, “Off-line Nepali handwritten character recognition using Multilayer Perceptron and Radial Basis Function neural networks,” *2012 Third Asian Himalayas International Conference on Internet*, Kathmundu, Nepal, pp. 1-5, 2012.
- [36] V. M. Lomte and D. D. Doye, “Handwritten Vedic Sanskrit Text Recognition Using Deep Learning,” *Journal of Algebraic Statistics*, vol. 13, no. 3, pp. 2190-2198, 2022.
- [37] U. Pal, M. Mitra and B. B. Chaudhuri, “Multi-skew detection of Indian script documents,” *Proceedings of Sixth International Conference on Document Analysis and Recognition*, Seattle, WA, USA, pp. 292-296, 2001.
- [38] S. V. Shelke and D. M. Chandwadkar, “Automatic System for Recognition of Handwritten

- Character using Multiscale Neural Network,” *International Journal of Computer Applications*, vol. 6, 2022.
- [39] S. Kubatur, M. Sid-Ahmed and M. Ahmadi, “A neural network approach to online Devanagari handwritten character recognition,” *2012 International Conference on High Performance Computing & Simulation (HPCS)*, Madrid, Spain, pp. 209-214, 2012.
- [40] S. V. Shelke and D. M. Chandwadkar, “Handwritten Devnagri Character Recognition,” *GRD Journal for Engineering*, vol. 1, no. 5, pp. 67–70, Apr. 2016.
- [41] K. U. Sharma and A. A. Ugwekar, “English transcription of sanskrit characters using predefined templates,” *2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, Coimbatore, India, pp. 1-4, 2017.
- [42] S. P. Patil and K. N. Jariwala, “Handwritten English, Marathi and Sanskrit Word Recognition Using Artificial Neural Network And Feature Extraction,” *International Journal of Advances in Science, Engineering and Technology (IJASEAT)*, vol. 6, no. 2, pp. 4–7, 2018.
- [43] V. More, M. Kharat and S. Gumaste, “Segmentation Of Devanagari Handwritten Text Using Feature Extraction,” *International Journal of Scientific & Technology Research*, vol. 9, no. 3, pp. 6594–6605, 2020.
- [44] J. Sachdeva and S. Mittal, “Handwritten Offline Devanagari Compound Character Recognition Using CNN,” in *Proceedings of Data Analytics and Management*, Springer, pp. 211-220, 2021.
- [45] N. Azawi, “Handwritten digits recognition using transfer learning,” *Computers and Electrical Engineering*, vol. 106, p. 108604, Jan. 2023.
- [46] P. P. Nair, A. James, P. Simon and P. V. Bhagyasree, “Malayalam Handwritten Character Recognition using CNN Architecture,” *Indonesian Journal of Electrical Engineering and Informatics*, vol. 11, no. 3, pp. 764–777, Sep. 2023.
- [47] S. V. Shelke, “Discrete wavelet transform and convolutional neural network based handwritten Sanskrit character recognition,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 38, no. 2, pp. 1367–1375, 2025.
- [48] A. Pradhan, “India to digitize scripts,” *Times of India*, May 9, 2014. [Online]. Available: <https://timesofindia.indiatimes.com/india/india-to-digitize-scripts/articleshow/34844112.cms>



Shraddha V. Shelke has completed her B.E and M.E. in Electronics and Telecommunication from University of Pune in 2009 and 2012 respectively. She is currently pursuing her Ph. D. in Electronics and Telecommunication from Savitribai Phule Pune University.



Pune University.

Dinesh M. Chandwadkar is a Professor in E & TC Department at K. K. Wagh Institute of Engineering Education & Research, Nashik, India. His area of interest includes Signal Processing, Power Electronics, Mechatronics, and Automotive Electronics etc. He has published More than 50 research papers in reputed Journals. He is working as Board of Studies member of Electronics and Telecommunication Engineering for



Sunita P. Ugale is working as a Professor and Head of Department in Electronics and Telecommunication Engineering department of K. K. Wagh Institute of engineering Education and Research, Nashik, Maharashtra. She has work experience of more than 30 years. Her special fields of interest include Fiber Optics Communication, Optical Sensors, Automation and VLSI technology.



Rupali V. Chothe is working as an Assistant Professor in Electronics and Telecommunication Engineering department of K. K. Wagh Institute of Engineering Education and research, Nashik, Maharashtra since last 17 years. She is pursuing Ph.D. from Savitribai Phule Pune University.