A Review on Devanagari OCR for Handwritten Text

Mamta Bisht[1,2,*] and Richa Gupta[2]

[1] *Department of Computer Science and Engineering (AI & ML), Inderprastha Engineering College, Ghaziabad, India*
[2] *Department of Electronics and Communication Engineering, Jaypee Institute of Information Technology, Noida, India*
E-mail: bishtmamta29@gmail.com; mamtabisht@ipec.org.in; richa.gupta@jiit.ac.in
Orcid id: https://orcid.org/0000-0002-5916-059X
[*] *Corresponding Author*

Received 29 November 2025; Accepted 06 February 2026

Abstract

With the rapid growth of document digitization and multilingual content, robust Optical Character Recognition (OCR) systems have become increasingly important. While substantial progress has been achieved for Latin scripts, accurate text detection and recognition in complex Indic scripts, particularly Devanagari, remain challenging due to script-specific structural characteristics, handwriting variations, and limited benchmark datasets. This review paper presents a comprehensive and structured analysis of existing research on text detection, script identification, and handwritten numeral, character, and word recognition for the Devanagari script. The paper systematically categorizes and compares conventional image processing methods, machine learning techniques, and modern deep learning-based approaches, highlighting their strengths and limitations. In addition, key challenges related to segmentation, multilingual scenarios, degraded documents, and

Journal of Graphic Era University, Vol. 14_1, 271–304.
doi: 10.13052/jgeu0975-1416.1419

resource constraints are critically discussed. By identifying open research gaps and outlining potential future research directions, this work aims to serve as a valuable reference and roadmap for researchers and practitioners working on Devanagari OCR and multilingual document analysis.

Keywords: Optical character recognition (OCR), Devanagari script, text detection, text segmentation, script identification, handwritten text recognition.

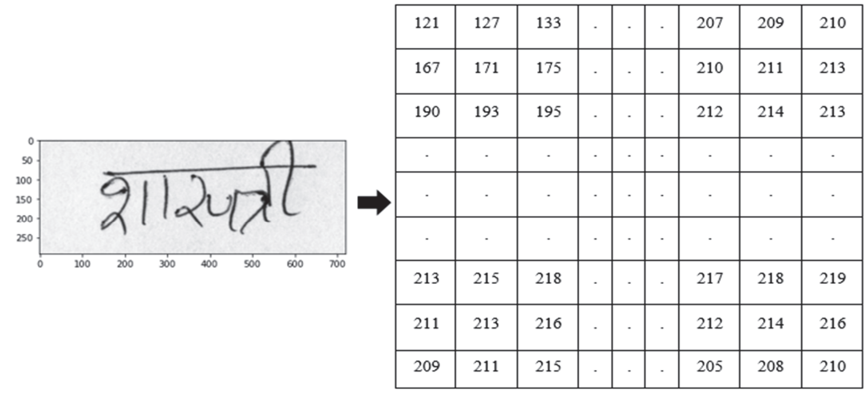
1 Introduction

The process of converting the text that exists in scanned documents or images into a computer-readable form that can subsequently be edited, searched, and used for further processing is known as Optical Character Recognition (OCR). In other words, OCR is a way to turn text in images into a format that a computer can read. This means that the text can be selected, edited, searched, etc. As depicted in Figure 1, the input image for a computer is merely a collection of pixel values. The computer is unable to determine whether an image contains text, a vehicle, an animal, etc. Therefore, if we have an image of text, it is unreadable by machines. In addition, we cannot select, edit, or perform any additional processing on this text in image format. OCR is used to convert this text into a machine-readable format. Basically, the OCR system digitizes printed text, which is the procedure of converting text images into a digital format that a computer can interpret. In other words, it takes this image as input and makes a text file with all the text in the image, as illustrated in Figure 2.

Learning OCR is important due to its wide range of real-life applications that simplify daily tasks. It enables automatic data entry from documents like cheques, passports, and invoices, supports number plate recognition and self-driving car navigation, and aids in book digitization. OCR also benefits visually impaired individuals through text-to-speech conversion.

The various challenges affecting the recognition performance of an OCR system in images captured by a camera or scanned by a scanner are outlined in the following points.

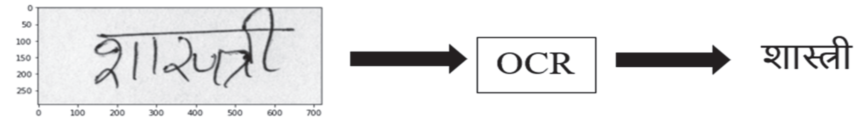
Noise in Input Images: Noise in input text images primarily originates from the document scanning process. Contributing factors include low image resolution and poor text contrast, which may result from scanner limitations



121	127	133	.	.	.	207	209	210
167	171	175	.	.	.	210	211	213
190	193	195	.	.	.	212	214	213
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
213	215	218	.	.	.	217	218	219
211	213	216	.	.	.	212	214	216
209	211	215	.	.	.	205	208	210

Text image Pixel values of input

Figure 1 Image as an array of pixel values.



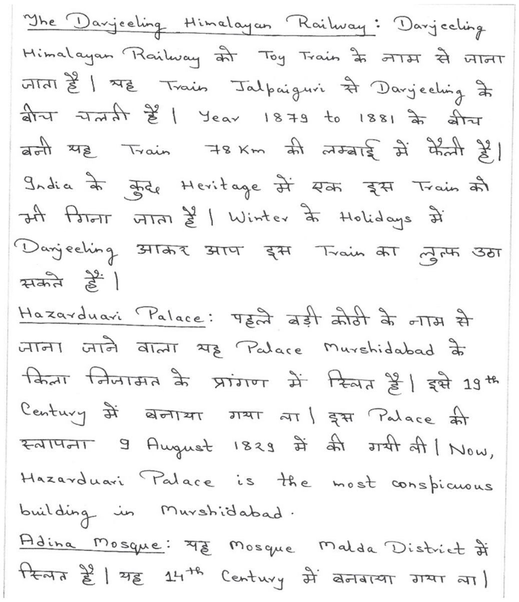
Input image Output text

Figure 2 Text image to editable text using OCR.

or the use of mobile phone scanning apps under suboptimal lighting conditions. Such noise can significantly degrade the performance of OCR models, making the application of noise reduction techniques a critical step in the image pre-processing stage.

Skew in Input Images: When the lines of a multi-line text image are slanted or inclined rather than horizontally aligned, it becomes difficult to accurately detect and segment the text lines. This issue is particularly pronounced in the case of the Devanagari script, where the presence of ascenders, descenders, and complex character structures further complicates the task of line detection and alignment during the recognition process.

Images Embedded with Text in Input Images: The presence of images interspersed with text in an input image introduces challenges for text recognition, as it complicates the task of accurately isolating and extracting the textual content.



The Darjeeling Himalayan Railway: Darjeeling Himalayan Railway को Toy Train के नाम से जाना जाता है। यह Train Jalpaiguri से Darjeeling के बीच चलती है। Year 1879 to 1881 के बीच बनी यह Train 78 Km की लम्बाई में फैली है। India के कुछ Heritage में एक इस Train को भी गिना जाता है। Winter के Holidays में Darjeeling आकर आप इस Train का लुत्फ उठा सकते हैं।

Hazarduari Palace: पहले बड़ी कोठी के नाम से जाना जाने वाला यह Palace Murshidabad के किला निजामत के प्रांगण में स्थित है। इसे 19th Century में बनाया गया था। इस Palace की स्थापना 9 August 1829 में की गयी थी। Now, Hazarduari Palace is the most conspicuous building in Murshidabad.

Adina Mosque: यह Mosque Malda District में स्थित है। यह 14th Century में बनवाया गया था।

Figure 3 Multilingual documents written in Devanagari and Latin.

Multilingual Text in Input Images: Input documents or scene images may contain a mixture of two or more languages, which adds complexity to the text recognition process. As illustrated in Figure 3, the document is primarily written in the Devanagari script, with occasional words in the Latin script interspersed. Similarly, the scene image shown in Figure 4 demonstrates a multilingual composition, featuring text written in four different scripts: Devanagari, Latin, Gurmukhi, and Urdu.

Text Miscellany: In both printed and handwritten documents, it is common to encounter a variety of fonts, typefaces, styles, and character sizes. Each variation in font or size can significantly alter the visual structure of a character, making it harder for recognition algorithms to accurately identify and interpret the text. Additionally, mixed formatting within a single document, such as bold or italicized text, decorative fonts, or irregular spacing, poses further challenges in segmenting and recognizing characters correctly during image processing. Addressing text miscellany is therefore a critical aspect of developing robust and adaptable OCR systems.

Post-processing: After the initial recognition phase in OCR, the output often contains errors or formatting issues, especially with complex scripts like

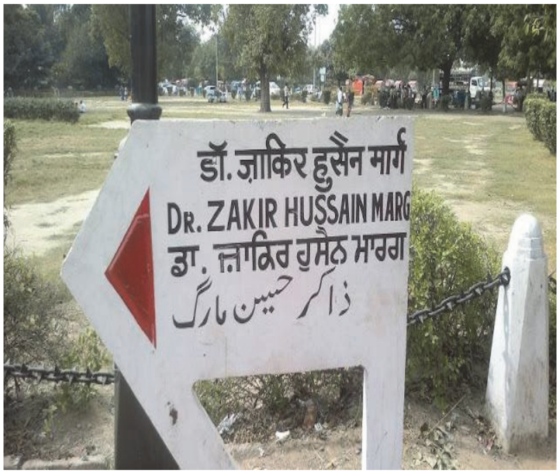


Figure 4 Multilingual scene text written in Devanagari, Latin, Gurumukhi, and Urdu.

Devanagari. To convert this raw recognized text into a coherent and accurate digital format, a significant amount of linguistic knowledge is essential. This includes understanding grammar, context, word boundaries, spelling correction, and script-specific nuances. Effective post-processing ensures that the final output is not only machine-readable but also meaningful and usable for further applications such as text-to-speech, translation, or search indexing.

These factors complicate accurate text detection and recognition, especially in complex scripts like Devanagari. Therefore, Devanagari script deserves special focus due to its structural complexity and widespread use across several major Indian languages, including Hindi, Marathi, Sanskrit, and Nepali. Its distinctive shirorekha (headline), large set of modifiers and compound characters, and high visual similarity among many handwritten characters make segmentation and recognition particularly challenging. Moreover, the scarcity of benchmark datasets and the need for accurate digitization of large volumes of handwritten and historical documents further underscore the importance of focused research on Devanagari OCR.

These challenges motivate the need for a focused review that critically examines existing solutions and identifies directions for improvement in Devanagari OCR. Although several studies address OCR for Indian scripts, a unified and systematic review focusing on text detection, script identification, and handwritten recognition challenges specific to the Devanagari script remains limited [1, 2]. In this context, the present review provides

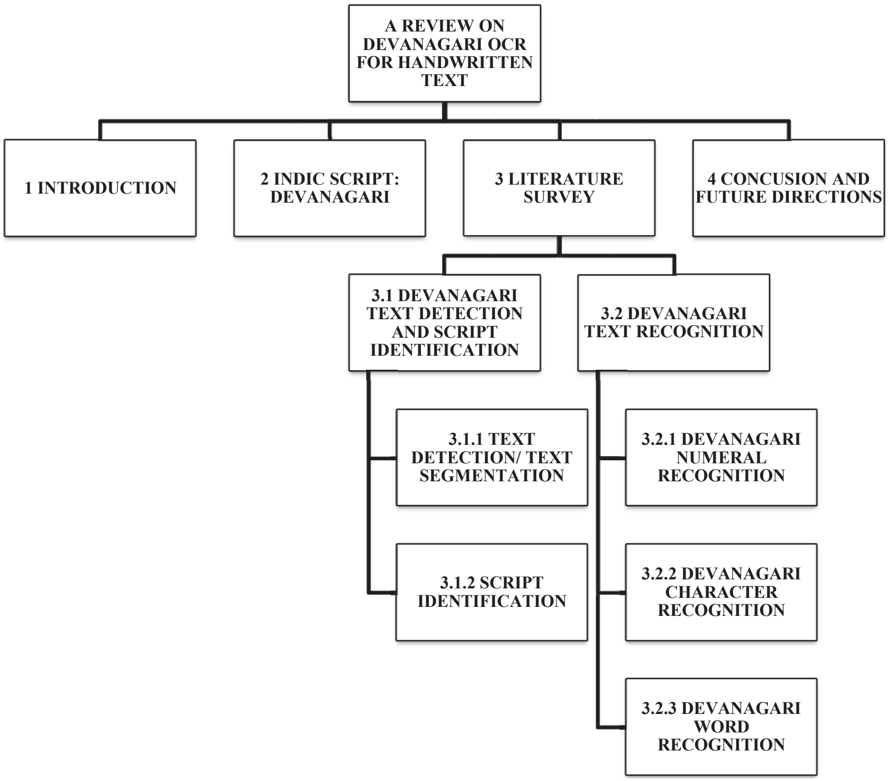

A REVIEW ON DEVANAGARI OCR FOR HANDWRITTEN TEXT
1 INTRODUCTION
2 INDIC SCRIPT: DEVANAGARI
3 LITERATURE SURVEY
4 CONCUSION AND FUTURE DIRECTIONS
3.1 DEVANAGARI TEXT DETECTION AND SCRIPT IDENTIFICATION
3.2 DEVANAGARI TEXT RECOGNITION
3.1.1 TEXT DETECTION/ TEXT SEGMENTATION
3.1.2 SCRIPT IDENTIFICATION
3.2.1 DEVANAGARI NUMERAL RECOGNITION
3.2.2 DEVANAGARI CHARACTER RECOGNITION
3.2.3 DEVANAGARI WORD RECOGNITION

Figure 5 The flow of review paper.

a comprehensive and structured synthesis of existing methods, highlights script-specific challenges, and identifies open research gaps and future directions. In Section 2 of this paper, the Devanagari script is described. A comprehensive review of relevant literature is provided in Section 3. Lastly, the conclusion and future directions of this paper are presented in Section 4. Figure 5 presents the complete flow of this review paper.

2 Indic Script: Devanagari

India is a linguistically diverse nation with a wide range of languages spoken across its population. Hindi, one of the official languages, is predominantly used in the central and northern regions and is written in the Devanagari script. In addition to Hindi, several other languages such as Marathi, Nepali, Sanskrit, and others also use this script. The Devanagari script comprises

Table 1 Devanagari characters

Swar (Vowels)	अ आ इ ई उ ऊ ए ऐ ओ औ अं अः ऋ
Vyanjan (Consonants)	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
Sanyukt vyanjan (Conjunct)	क्ष (=क्+ष), त्र (=त्+र), ज्ञ (=ज्+ञ), श्र (श्+र)
Barakhadi for vyanjan ख (Kha))	ख खा खि खी खु खू खे खै खो खौ खं खः
Few examples of sanyuktakshar (Conjunct)	क्+ र= क्र, त्+ व= त्व
Few examples of dwitiya vyanjan (Conjunct)	क्+ क = क्क, च्+ च= च्च,
Devanagari numbers	० १ २ ३ ४ ५ ६ ७ ८ ९
Matras (Modifiers)	◌ा ◌ि ◌ी ◌ु ◌ू ◌े ◌ै ◌ो ◌ौ ◌ं ◌ः ◌ृ ◌ँ ◌् ◌़

13 swars (स्वर) or vowels, 33 vyanjans (व्यंजन) or consonants, and 4 sanyukt vyanjans (संयुक्त व्यंजन), which are formed by combining two consonants, as shown in Table 1. When a consonant is combined with a vowel, its shape is modified at the top, bottom, left, or right, creating what is known as a modified character. For example, Table 1 presents the sequence of modified shapes for the consonant 'ख (Kha)', commonly referred to as Barakhadi (बारहखड़ी). The script also supports the formation of conjuncts, where two characters merge to form a new shape. When the same consonant is repeated, the resulting form is called a dwitiya vyanjan (द्वित्व व्यंजन), and when different consonants are combined, it is known as a sanyuktakshar (संयुक्ताक्षर). Table 1 also includes examples of these, along with the Devanagari numeral set and matras (modifiers).

The Devanagari script is written from left to right, with each character typically featuring a prominent horizontal line at the top known as the "Shirorekha" or headline. This line connects individual characters to form complete words. Segmenting text images into individual characters is particularly challenging in Devanagari due to the presence of conjuncts or joint characters such as क्क, क्ख, क्र, and त्र्य. Additional segmentation difficulties arise from the presence of isolated diacritical marks like Anuswar (◌ं), Visarga (◌ः), and Chandra Bindu (◌ँ), which serve as vowel modifiers. The recognition of ascenders and descenders also adds to the complexity. Moreover, the inclusion of bilingual or multilingual text in documents further complicates accurate text recognition in the Devanagari script.

3 Literature Survey

The goal of developing a text recognition system is to enable machines to read, edit, and interact with text in a manner similar to humans. Advancements in OCR have direct practical applications in the digitization of historical manuscripts, automated processing of government and legal records, postal and banking systems, educational document analysis, and multilingual information retrieval, thereby supporting large-scale digital archiving and e-governance initiatives. This field has attracted significant attention from researchers, leading to numerous advancements [3]. Despite this progress, handwriting recognition remains a challenging task due to the diversity of handwriting styles, the presence of visually similar characters, and the vast number of character classes particularly in Indian scripts. While existing OCR software performs well on printed text, it often struggles with handwritten characters, especially in scripts like Devanagari that exhibit considerable variation in character sets. To address these challenges, a wide range of traditional techniques and modern deep learning approaches have been proposed in the literature. Nevertheless, improving OCR performance for handwritten text remains an active area of research, motivating continued efforts in this domain.

This section presents an in-depth overview of OCR systems developed for Devanagari text documents. The process begins with the optical scanning of documents using either a scanner or a digital camera. This is followed by text detection and segmentation in the scanned document or natural scene image. Subsequently, character recognition is performed through various stages of the OCR pipeline. Section 3.1 discusses research related to text detection and script identification. Section 3.2 delves into studies focused on numeral, character, word, and line recognition.

3.1 Devanagari Text Detection and Script Identification

Text detection and script recognition pose fundamental challenges when dealing with text images, whether they are found in documents or in scene text. Text detection aims to accurately locate any textual content within an image, usually represented by bounding boxes. On the other hand, script recognition emphases on determining the specific script utilized to write the text present in an image.

The following sections detail the research conducted on text detection and script identification:

3.1.1 Text detection/text segmentation

OCR systems primarily focus on identifying and processing text regions. A document page or a natural scene image may contain both text and non-text regions. In such cases, the ability to accurately detect and separate text from non-text areas is crucial a process known as page segmentation [4]. Line segmentation involves identifying and isolating individual lines of text from a page that contains multiple lines. This task is relatively straightforward for printed or neatly handwritten documents but becomes challenging when dealing with skewed text, irregular spacing between lines, or varying text sizes. Similarly, word segmentation refers to dividing each line into individual words, while character segmentation involves isolating individual characters within each word. When the gaps between words and characters are minimal, segmentation becomes significantly more difficult [5–7]. As a result, modern OCR approaches increasingly prioritize word and line detection over character detection, which tends to be slower and more complex.

The previous work on text detection can be generally divided into two categories: conventional approaches and approaches based on deep learning. Conventional methods involve manual feature extraction to differentiate text from the background [8–13]. Popular conventional methods, such as Stroke Width Transform (SWT) [8] and Maximally Stable Extremal Regions (MSER) [7], are commonly employed for manually designing features in text detection tasks. Typically, these approaches extract candidate characters by means of edge detection or extracting extreme regions. Zhang et al. [14] also consider the local symmetry property of text in natural scene images while developing features for text detection purposes. Busta et al. [15] focused on rapid text detection and enhanced the FAST keypoint detector for stroke extraction. In their work, Dhar et al. [16] introduced a gradient morphology-based approach for detecting multilingual scene text.

In contrast, deep learning-based techniques extract effective attributes from training datasets during the learning process [17–24] and recently, these techniques have shown significant performance for text detection problems [1, 23, 24]. Deep learning-based text detection techniques can generally be classified into two categories: multi-step methods and simplified (single-step) methods. A comparison of several recent text detection approaches for scene images is illustrated in Figures 6 and 7, where each box represents an individual processing step. As shown in Figure 6, the three algorithms (a)

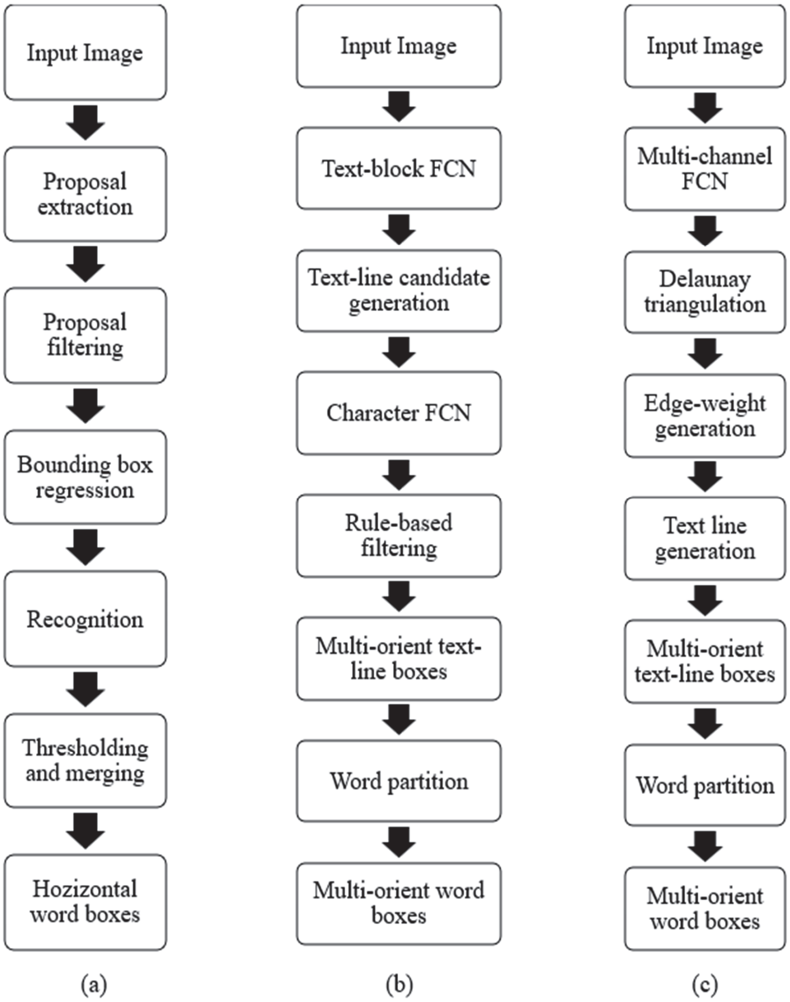


Figure 6 Text detection methods in deep learning with multi-steps.

by Jaderberg et al. [20], (b) by Zhang et al. [22], and (c) by Yao et al. [28] consist of multiple stages and therefore fall under the category of multi-step approaches.

On the other hand, the two algorithms shown in Figure 7, proposed by Tian et al. [24] and Zhou et al. [23], follow simplified processing steps. In Figure 7, algorithm (a) refers to the Connectionist Text Proposal Network (CTPN), while algorithm (b) refers to Efficient and Accurate Scene Text

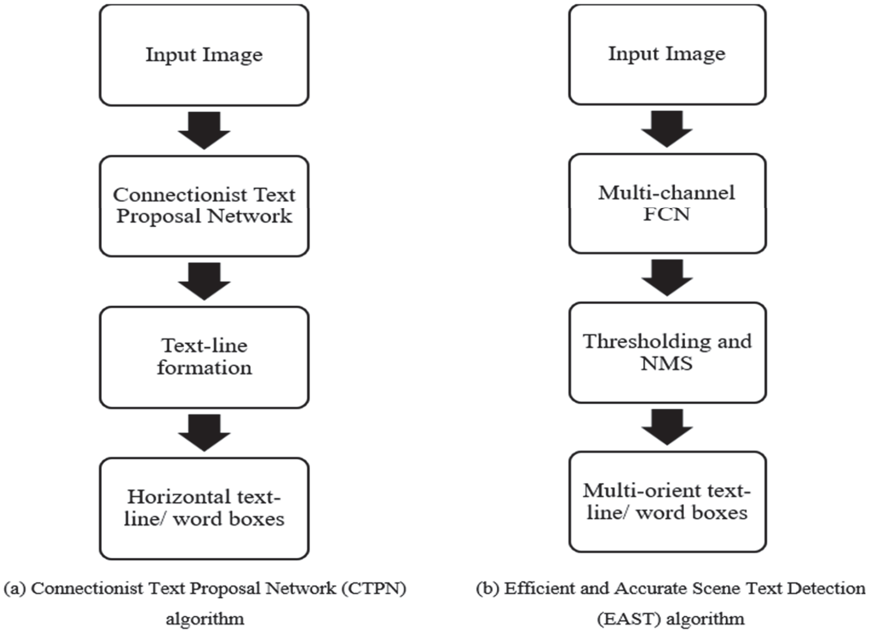
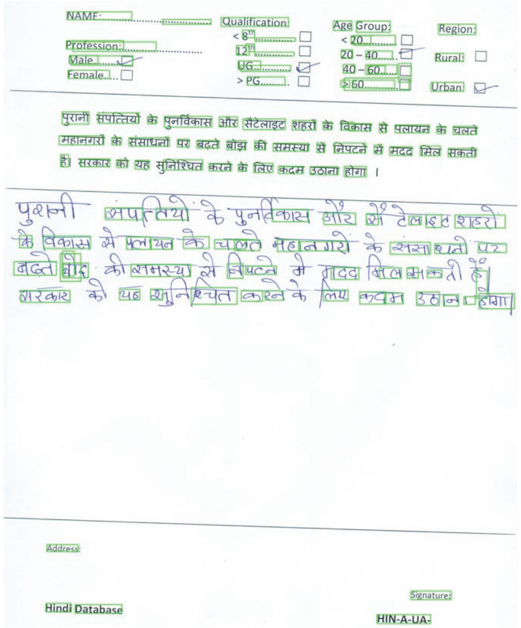

Input Image
Connectionist Text Proposal Network
Text-line formation
Horizontal text-line/ word boxes
(a) Connectionist Text Proposal Network (CTPN) algorithm
Input Image
Multi-channel FCN
Thresholding and NMS
Multi-orient text-line/ word boxes
(b) Efficient and Accurate Scene Text Detection (EAST) algorithm

Figure 7 Text detection methods in deep learning with simplified steps.

Detection (EAST), both designed for natural scene text detection. These widely recognized algorithms streamline the detection process by reducing intermediate steps, making them more efficient compared to the three multi-step methods illustrated in Figure 6. In CTPN, the input image is fed into the model, which generates fine-scale text proposals. These proposals are then linked using a specific algorithm to produce the final detected text regions. The EAST algorithm consists of two main stages. First, the input is passed through a multi-channel Fully Convolutional Network (FCN), which directly outputs score and geometry maps. These maps are then converted into text regions, followed by post-processing techniques such as thresholding and Non-Maximum Suppression (NMS) to generate the final text bounding boxes.

Existing deep learning methods for text detection and localization demonstrate satisfactory results with printed text, but their effectiveness diminishes when confronted with handwritten text within an image. Additionally, modern deep learning techniques rely on extensive datasets for training, which poses challenges when the available dataset and computational resources are limited. To address this issue, Bisht and Gupta [29] suggested the utilization of pre-trained models as a solution. Bisht and Gupta [30] also



NAME:..

Profession:..

Male......☑

Female.....□

Qualification

< 8th.............. □

12th.............. □

UG.............. ☑

> PG.............. □

Age Group:

< 20.............. □

20 – 40.............. ☑

40 – 60.............. □

≥ 60.............. □

Region:

Rural: □

Urban: ☑

पुरानी संपत्तियों के पुनर्विकास और सैटेलाइट शहरों के विकास से पलायन के चलते महानगरों के संसाधनों पर बढ़ते बोझ की समस्या से निपटने में मदद मिल सकती है। सरकार को यह सुनिश्चित करने के लिए कदम उठाना होगा ।

पुरानी सपत्तियो के पुनर्विकास और सैटेलाइट शहरो के विकास से पलायन के चलते महानगरो के ससाधनो पर बढ्ते बोझ की समस्या से निपटने मे मदद मिल सकती है। सरकार को यह सुनिश्चित करने के लिए कदम उठाना होगा।

Address:

Signature:

Hindi Database

HIN-A-UA-

Figure 8 Text detection and localization using pre-trained EAST model for handwritten text in Devanagari script.

evaluated the performance of a well-known pre-trained EAST model in their study on handwritten documents in the Devanagari script, as depicted in Figure 8. It is clear from Figure 8 that the pre-trained EAST model works well for printed parts of text, but the performance is poor for handwritten text. They further applied image processing and an effective morphology-based technique for text detection and localization for small handwritten text images in their work.

3.1.2 Script identification

The text detection and segmentation phases are responsible for isolating meaningful text components from an image. In multilingual scenarios, the next crucial step is script identification, which plays a vital role in OCR systems, as its accuracy significantly influences the performance of subsequent recognition stages. Script identification methods can broadly be categorized into machine learning approaches [31, 32] and deep learning approaches [33–35]. Earlier research in this domain primarily focused on the structural and visual characteristics of scripts in documents [36, 37] and videos [38, 39].

Traditional methods involved extracting handcrafted features from the data, which were then used to train classifiers for script recognition. Commonly employed classifiers included Support Vector Machines (SVM), K-Nearest Neighbors (K-NN), and Random Forest (RF). In recent years, Convolutional Neural Networks (CNNs) have gained considerable attention for their ability to automatically extract meaningful features from raw input data [40–42], reducing the reliance on manual feature engineering—an area where conventional machine learning techniques often struggle due to high time and labor demands.

In the literature, Mei et al. [43] proposed a script identification framework using an end-to-end trainable network based on CNN and RNN, incorporating an average pooling mechanism to handle input images of varying dimensions. Shi et al. [34] developed a CNN model employing multistage pooling, allowing it to process scene text images with highly variable aspect ratios. Their subsequent work [35] focused on extracting intermediate representations and utilizing them within CNNs for script classification. Gomez and Karatzas [33] extracted CNN features before applying a Naive Bayes classifier, while Bhunia et al. [44] also leveraged CNN features, later enhancing their method with an attention mechanism.

Lu et al. [45] introduced a hybrid approach that combined local and global CNNs along with the ResNet-20 architecture for script recognition. Ma et al. [46] proposed a method using hierarchical feature fusion in conjunction with CNNs, and Tounsi et al. [47] applied transfer learning techniques to classify scripts using a relatively small dataset.

3.2 Devanagari Text Recognition

A comprehensive review of existing machine learning and deep learning approaches for handwritten numeral, character, and word/line recognition in the Devanagari script is presented in the following section

3.2.1 Devanagari numeral recognition

The literature encompasses various approaches for feature extraction and categorization. In their work on Devanagari numeral recognition, Bajaj et al. [48] focused on statistical features. They introduced density features as the first statistical feature, capturing the shape of numerals through the distribution of pixel densities. Additionally, they computed kurtosis and skewness as the second statistical feature, representing the flatness and asymmetry of the distribution. Normalized skewness and kurtosis were also calculated to measure the relationship between symmetry and flatness in the distribution. Third, they constructed the descriptive features for the Devanagari numerals in terms of different types of strokes. Elnagar and Harous [49] focused on structural aspects of thinned cursive handwritten Hindi numerals, such as strokes and cavity features. Ramteke and Mehrotra [50] have divided the numerals into systematic parts, and features from each divided part were calculated. These characteristics are constructed utilizing the notion of invariant moments. They also used Principal Component Analysis (PCA) to balance the pixel spreading in all the regions of the divided image. Their work increases the numeric recognition rate over the correlation coefficient method. A study conducted by Garain et al. [51] aimed to evaluate the effectiveness of a clonal selection approach in categorizing handwritten numerals into 10 distinct classes. Classification was accomplished by Sharma et al. [52] utilizing a quadratic classifier and 64-dimensional features that were derived from the chain code histogram. After segmenting the numerals and characters into blocks, they measured the chain code histogram information for every block separately. Hanmandlu and Murthy [53] have shown that the numerals can be modelled as modified exponential relationship functions that can then be converted to fuzzy sets. In order to generate normalized distance attributes, the box technique is utilized. These features are then incorporated into a classifier. To extract features of handwritten Hindi numerals, Hanmandlu et al. [54] also used a box approach, which they followed up with a bacterial foraging strategy. For their research, Patil and Sontakke [55] looked into a fuzzy Neural Network (NN) classifier for Devanagari numerals, taking into account their structural properties. Pal et al. [56] extracted orientation data from the contour points of handwritten digits and applied a Modified Quadratic Discriminant Function (MQDF) for classification purposes. A multistage cascading approach strategy was presented by Bhattacharya and Chaudhuri [57]. This strategy places an emphasis on wavelet-based multi-resolution features as well as Multi-Layer Perceptron (MLP) classifiers. In their work on postal automation, Pal et al. [58] made use of gradient features and the MQDF classifier. Basu et al. [59] employed a Quad-Tree-Based Longest Run (QTLR) technique to extract attributes from numeric digit designs. They evaluated the extracted features using a SVM classifier. With the help of Fourier descriptors, Rajput and Mali [60] were able to identify individual Marathi handwritten digits as features, which they then used for three distinct classifiers: nearest neighbor, K-nearest neighborhood (K-NN), and SVM. The Deep CNN for Devanagari handwritten characters, including numerals, was given by Acharya et al. [61]. Khanduja et al. [62] introduced a methodology for extracting statistical and structural features for numeric and character recognition in Devanagari script. They employed a multilayer perceptron (MLP) classifier for their recognition tasks. Bisht and Gupta [63] also presented a CNN-based model with SGD optimizer for classification of characters, including numerals.

3.2.2 Devanagari character recognition

For character recognition in Devanagari script, Arora et al. [64] focused on structural features of characters, emphasizing the headline (shirorekha), spine, and junction points. These features were then classified using a feed-forward neural network (NN). Hanmandlu et al. [65] proposed a vector distance-based feature extraction method combined with a fuzzy set classifier for character classification. Pal et al. [66] employed gradient and Gaussian filters for feature extraction and used a quadratic classifier for recognition. Deshpande et al. [67] utilized chain code-derived features and applied techniques such as regular expressions and the Minimum Edit Distance (MED) scheme for recognition. In their work, Pal et al. [68] categorized features into directional and curvature-based types. Directional information was computed using the arc tangent of gradients, while curvature-based features also relied on gradient data. Classification was carried out using SVM and Modified Quadratic Discriminant Function (MQDF) classifiers. Arora et al. [69] presented a combination of feature extraction and recognition methods, where extracted features were initially classified using a Multi-Layer Perceptron (MLP), and the outputs of multiple MLPs were combined using a weighted majority voting scheme.

Kumar [70] evaluated five feature extraction techniques on a handwritten Devanagari dataset and found that SVM performed best when trained on gradient-based features. Mane and Ragha [71] introduced an eigen-deformation-based elastic image matching technique. They modeled category-specific deformations (eigen-deformations) as feature sets and applied elastic matching for character recognition. Pal et al. [72] conducted a comparative study involving twelve classifiers and four types of features based on curvature and gradient information. Their findings highlighted the Mirror Image Learning (MIL) classifier as the most effective. Arora et al. [73] proposed a two-stage character recognition system: the first stage utilized two MLPs – one for shadow features and the other for chain code histograms – and combined their outputs using a weighted majority rule. In the second stage, relative difference values from the MLP outputs were used to group characters into distinct forms and similar-looking characters, which were then classified using MLP and MED, respectively.

Puri and Singh [74] applied a projection profile method for document segmentation into lines, words, and characters. They further removed the shirorekha from characters to focus on their geometric structures and used these modified characters to train an SVM classifier. Deore and Pravin [75] employed a fine-tuning strategy using a pre-trained VGG-16 model for feature extraction in handwritten character recognition. Pande et al. [76] developed a CNN-based architecture with dropout layers for improved recognition performance. Additionally, a study [77] explored modified character recognition using a CNN model. Beyond recognition, the generation of handwritten Devanagari numerals and consonants has also been investigated using Conditional GAN architectures [78].

A summary of the literature-based research on Devanagari handwritten numeral and character recognition is illustrated in Figures 9 and 10.

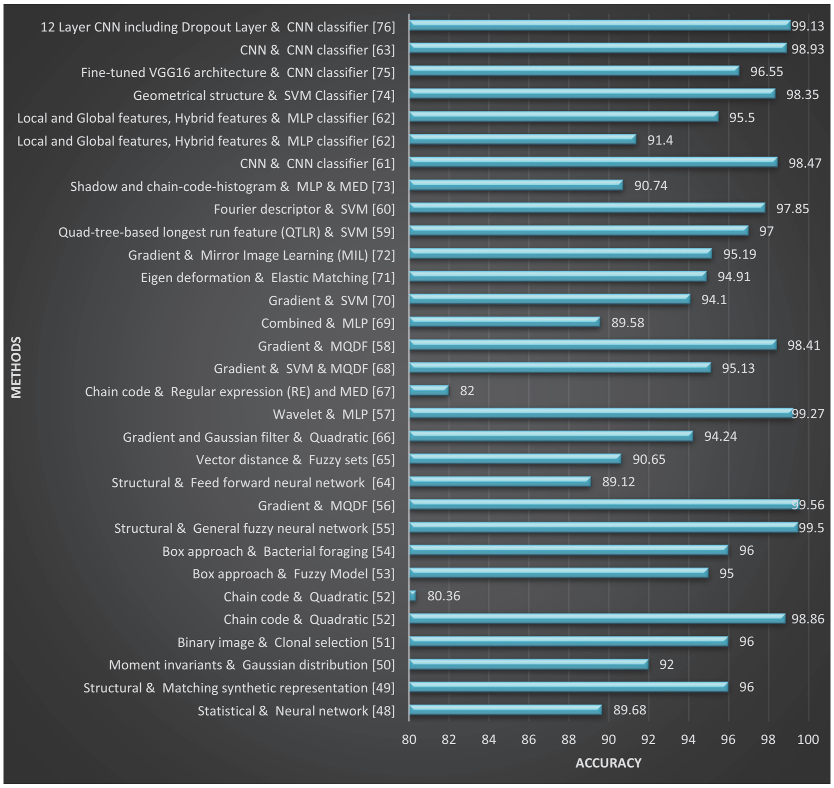


Figure 9 Summary of significant research for Devanagari character and numeral recognition: Method vs Accuracy.

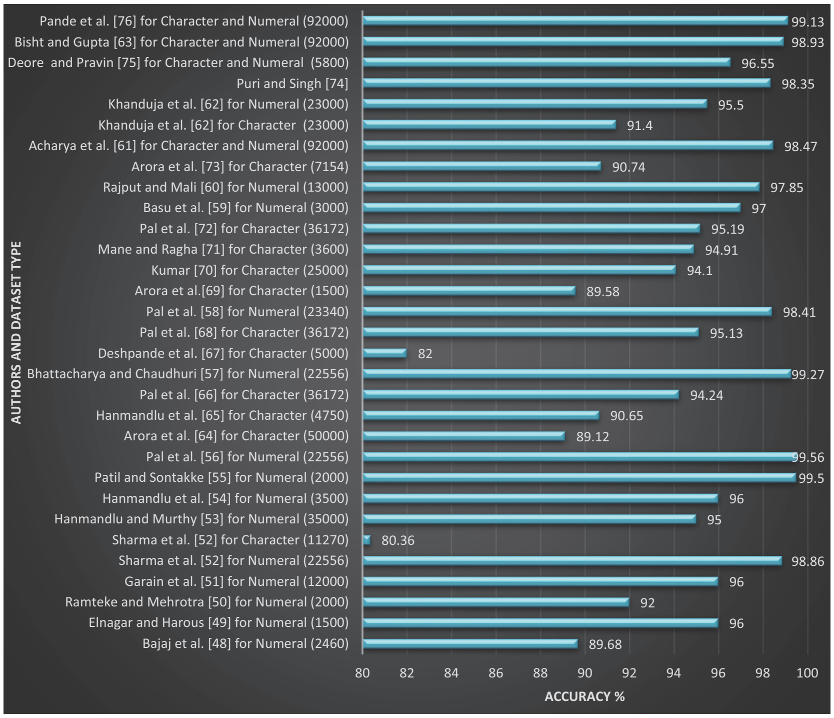


Figure 10 Summary of significant research for Devanagari character and numeral recognition: Dataset type and size vs Accuracy.

These figures showcase different feature extraction and classification methods, along with the recognition accuracy achieved for handwritten numerals and characters in Devanagari script. The findings reveal that the literature encompasses a range of approaches for Devanagari handwritten numeral and character recognition, demonstrating promising performance. It is also observed that for large datasets, the CNN-based method outperforms. The CNN-based method also makes it easier to get feature vectors for further classification steps without having to do complicated manual calculations.

3.2.3 Devanagari word recognition

The literature on Devanagari handwritten word recognition can broadly be categorized into two main approaches: segmentation-based methods [79] and holistic-based methods [80–82]. Segmentation-based methods involve breaking down a handwritten word into individual characters, followed by applying character recognition techniques using feature extraction and classification.

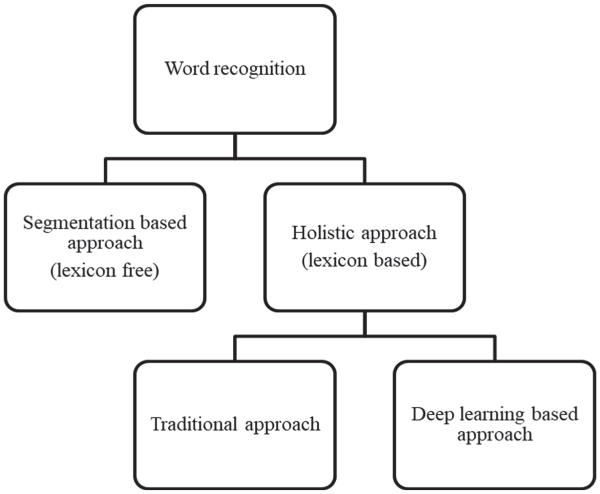

Word recognition
Segmentation based approach (lexicon free)
Holistic approach (lexicon based)
Traditional approach
Deep learning based approach

Figure 11 Word recognition methods.

The recognized characters are then concatenated to form the final word. However, the effectiveness of this approach heavily depends on accurate segmentation, which poses challenges in cases of touching or overlapping characters and noisy images. To overcome these limitations, researchers have increasingly turned to holistic-based methods for recognizing entire handwritten words without explicit character segmentation. As illustrated in Figure 11, holistic-based approaches can be further divided into two categories: traditional techniques and modern deep learning-based methods.

Parui and Shaw [83] proposed a segmentation-free strategy for recognizing handwritten words, leveraging Hidden Markov Models (HMMs) and stroke-based features derived from a particular mixing distribution. In a related study, Shaw et al. [84] investigated the effectiveness of using histograms of chain code directions as feature vectors. These features were extracted from sequential image strips produced via a left-to-right sliding window, and recognition was performed using a continuous density HMM framework. Later, Shaw et al. [85] explored an alternative recognition pipeline based on explicit segmentation of handwritten words.

Singh et al. [86] developed a technique that applied curvelet-based feature extraction in combination with Support Vector Machines (SVM) and K-Nearest Neighbors (K-NN) classifiers to recognize handwritten Devanagari words. Meanwhile, Ramachandrula et al. [87] designed a holistic recognition method for Hindi handwritten words. Their approach involved generating a predictive character model, from which they extracted directional component features. Their reported recognition accuracy varied between 91.23% and 79.94%, depending on vocabulary size, ranging from 10 to 30 words.

In the domain of Devanagari handwritten town name recognition, Shaw et al. [88] and Shaw et al. [89] conducted two separate investigations. The first introduced an integrated method utilizing both contour and skeleton-based descriptors, classified using a multiclass SVM. The follow-up work concentrated on structural features derived from directional and gradient-based information, again employing a multiclass SVM for recognition.

Oval and Shirawale [90] presented a segmentation-free word recognition method by adopting a Recurrent Neural Network (RNN)-based architecture, effectively bypassing the need for explicit character segmentation.

Kumar [91] explored a segmentation-based technique for handwritten word recognition and reported accuracy levels across various word lengths. His findings indicated an 80.80% recognition rate for two-letter words and a 72% rate for six-letter words. He also highlighted several implementation challenges associated with the method. In a separate study, Roy et al. [92] proposed a recognition framework for offline handwritten Devanagari words. Their technique divided each word into three horizontal zones, with separate recognition performed for each zone. They used Pyramid Histogram of Oriented Gradient (PHOG) features, applying Hidden Markov Models (HMM) to the central zone and Support Vector Machines (SVM) to the upper and lower zones. The final word recognition result was obtained by integrating the outputs from all zones. Meanwhile, Malakar et al. [80] developed a detailed strategy for recognizing handwritten Hindi words. Their method focused on extracting a variety of features either from the full image or from its hypothesized sub-images. Each word was represented using an 89-dimensional feature vector. After evaluating multiple classifiers, they identified the Multi-Layer Perceptron (MLP) as yielding the best performance.

Dutta et al. [93] proposed a hybrid neural network model aimed at recognizing offline handwritten Devanagari words. Their architecture integrates Convolutional Neural Networks (CNN) with Recurrent Neural Networks (RNN), topped with a Connectionist Temporal Classification (CTC) layer. To further improve recognition accuracy, especially on handwritten datasets, they incorporated a Spatial Transformer Network (STN) within the CNN framework. Dwivedi et al. [94], on the other hand, introduced a new dataset comprising line-level annotated Sanskrit documents and developed an attention-based LSTM architecture for reading and analyzing such content. In a different study, Roy et al. [95] designed a CNN-driven approach to recognize handwritten Indian city names across various scripts. Their experiments spanned both single-script and multi-script environments to evaluate the robustness of their model. Bisht and Gupta [96] developed a comprehensive neural network system that automates the entire text recognition workflow. The system uses CNN layers to extract visual features, followed by LSTM layers to model sequential dependencies in the text. A transcription layer is also included for generating final predictions. To refine recognition outcomes, the researchers applied decoding strategies and performed lexicon correction. Their method was tested on two datasets featuring Devanagari handwritten words. On the first dataset, containing 95,381 samples, they achieved accuracies of 88.90% and 90.73% for lexicon sizes of 10K and 6K, respectively. On the second dataset, with 16,128 samples, accuracies of 80.95% and 91.28%

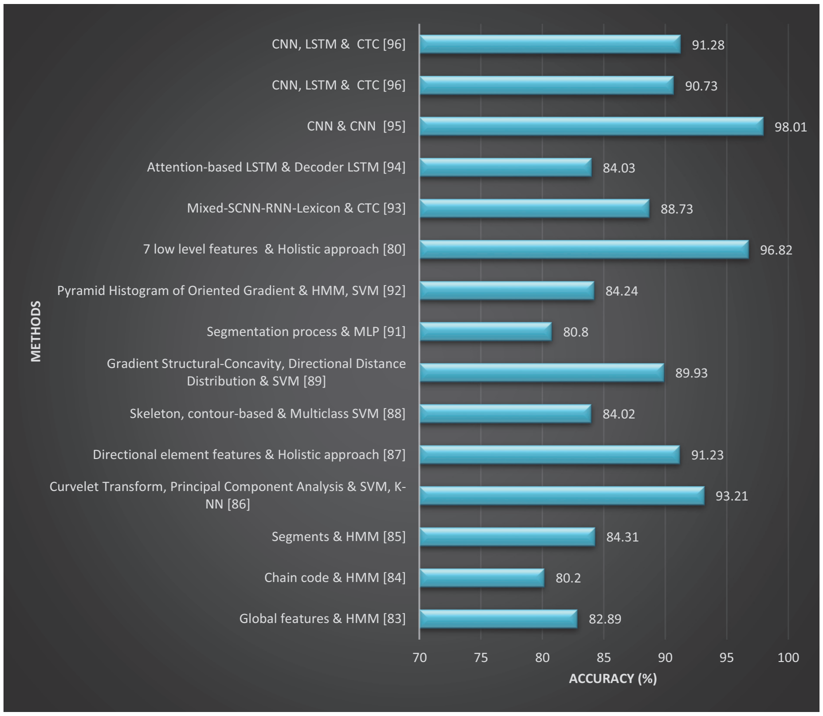


Figure 12 Summary of significant research for Devanagari word recognition: Method vs Accuracy.

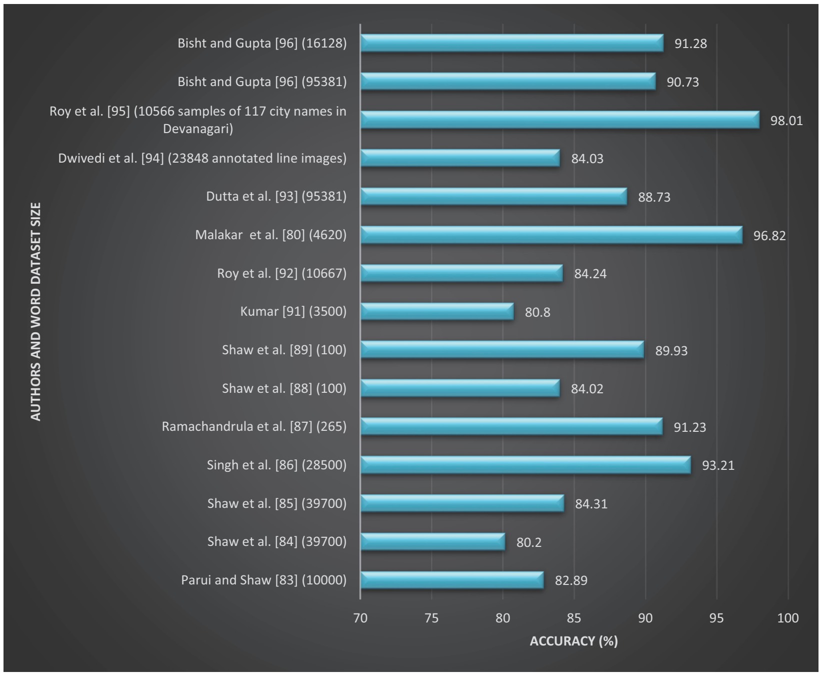


Figure 13 Summary of significant research for Devanagari word recognition: Size vs Accuracy.

were recorded for lexicon sizes of 2K and 1K, respectively. The authors in [97] present an end-to-end framework for Page-Level HTR by treating it as a two-stage problem involving word-level detection followed by recognition for handwritten text. The study examines six distinct HTR models, including three CRNN-based models (CRNN-VGG, MobileNet, and SAR) and three transformer-based models (MASTER, ViTSTR, and PARSEQ) on ten diverse Indic languages.

Figure 12 presents a comparison of various offline methods for recognizing handwritten Devanagari words at the word level. It showcases different feature extraction techniques, classification approaches, and highlights the highest achieved word recognition accuracy for Devanagari handwritten text. On the other hand, Figure 13 illustrates a performance comparison based on dataset size, incorporating insights from different authors in the literature.

A comparative table summarizing Devanagari OCR techniques along with their strengths and weaknesses is presented in Table 2.

Table 2 Comparison of major OCR techniques for Devanagari script

Approach	Techniques Used	Strengths	Limitations
Statistical/ Structural Methods	Zoning, density features, projection profiles, strokes, contours	Simple to implement, low computational cost, effective for small datasets	Sensitive to noise and handwriting variations; limited scalability
Handcrafted Feature + ML Classifiers	Gradient, chain code, curvature features with SVM, K-NN, MQDF, MLP	Good performance on isolated characters; interpretable features	Requires careful feature design; segmentation-dependent
Segmentation-Based OCR	Character-level segmentation + classifier	Works reasonably for clean, printed text	Poor performance for handwritten Devanagari due to shirorekha and touching characters
Holistic Word Recognition	HMM, sliding window features	Avoids character segmentation errors	Limited to fixed vocabularies; poor generalization to unseen words
Deep Learning (CNN-based)	CNN, CRNN, CNN–LSTM	Automatic feature learning; strong performance on large datasets	Data-hungry; high computational requirements
End-to-End DL Models	CNN–RNN–CTC, Attention, Transformers	Segmentation-free; handles variability well	Requires large labeled datasets; complex training

4 Conclusion and Future Directions

This review paper provides a concise yet comprehensive synthesis of research on text detection, script identification, and handwritten numeral, character, and word recognition for the Devanagari script. It systematically organizes existing literature across traditional, machine learning, and deep learning approaches, enabling clear comparison and understanding of methodological advancements. By explicitly identifying script-specific challenges, performance limitations, and unresolved research gaps, the paper offers meaningful insights beyond a conventional survey. The inclusion of structured summaries and comparative analyses enhances its usefulness as a reference resource. Moreover, the clearly defined future research directions make this work particularly valuable for guiding ongoing and future studies in Devanagari OCR and multilingual document analysis. The key future research directions are outlined as follows:

[1] **Robust segmentation-free text detection and recognition:** Develop end-to-end deep learning models capable of detecting and recognizing handwritten, curved, skewed, and non-horizontal Devanagari text, minimizing dependency on explicit character-level segmentation.

[2] **Benchmark datasets and evaluation standards:** Create large-scale, publicly available Devanagari handwritten datasets with standardized evaluation protocols to support fair comparison and reproducible research.

[3] **Synthetic data generation and augmentation:** Investigate GAN-based synthetic data generation and script-specific augmentation strategies to address data scarcity and improve model generalization.

[4] **Efficient transfer learning models:** Explore lightweight, transfer learning-based architectures with fewer trainable parameters to reduce computational cost while maintaining high recognition accuracy.

[5] **Recognition of degraded and historical documents:** Develop robust methods to handle degraded, noisy, and historical handwritten Devanagari documents, including ink bleed-through, faded strokes, and uneven backgrounds.

Declarations

Funding

Manuscript has no associated funding.

Author Contribution

Mamta Bisht: Methodology, Writing- Original draft preparation, Investigation, Survey.

Richa Gupta: Writing- Reviewing and Editing, Visualization, Supervision.

Conflict of Interest

The authors declare no conflict of interest.

Ethical Approval

The paper does not deal with any ethical problems.

Informed Consent

We declare that all the authors have informed consent.

Data Availability Statement

Manuscript has no associated data.

References

[1] S. Singh, N. K. Garg, and M. Kumar, "Feature extraction and classification techniques for handwritten Devanagari text recognition: a survey," *Multimedia Tools and Applications*, vol. 82, pp. 747–775, 2023.

[2] M. Jabde, C. H. Patil, A. D. Vibhute, and J. R. Saini, "A systematic review of multilingual numeral recognition systems," *Artif Intell Rev*, vol. 58, no. 4, p. 106, Jan. 2025, doi: 10.1007/s10462-025-11105-0.

[3] R. Plamondon and S. N. Srihari, "Online and off-line handwriting recognition: a comprehensive survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 63–84, 2000.

[4] S. Rani, "Recognition of Gurmukhi handwritten manuscripts," 2015.

[5] A. S. Ramteke and M. E. Rane, "Offline handwritten devanagari script segmentation," *International Journal of Scientific & Technology Research*, vol. 1, no. 4, pp. 142–145, 2012.

[6] R. Smith, D. Antonova, and D.-S. Lee, "Adapting the Tesseract open source OCR engine for multilingual OCR," in *Proceedings of the International Workshop on Multilingual OCR*, 2009, pp. 1–8.

[7] S. R. Narang, M. K. Jindal, and M. Kumar, "Drop flow method: an iterative algorithm for complete segmentation of Devanagari ancient manuscripts," *Multimedia Tools and Applications*, vol. 78, no. 16, pp. 23255–23280, 2019.

[8] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, 2010, pp. 2963–2970.

[9] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," in *Asian Conference on Computer Vision*, Springer, 2010, pp. 770–783.

[10] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2012, pp. 1083–1090.

[11] W. Huang, Z. Lin, J. Yang, and J. Wang, "Text localization in natural images using stroke feature transform and text covariance descriptors," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1241–1248.

[12] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2012, pp. 3538–3545.

[13] X.-C. Yin, X. Yin, K. Huang, and H.-W. Hao, "Robust text detection in natural scene images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 5, pp. 970–983, 2013.

[14] Z. Zhang, W. Shen, C. Yao, and X. Bai, "Symmetry-based text line detection in natural scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2558–2567.

[15] M. Busta, L. Neumann, and J. Matas, "Fastext: Efficient unconstrained scene text detector," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1206–1214.

[16] D. Dhar et al., "Multilingual scene text detection using gradient morphology," *International Journal of Computer Vision and Image Processing (IJCVIP)*, vol. 10, no. 3, pp. 31–43, 2020.

[17] A. Coates et al., "Text detection and character recognition in scene images with unsupervised feature learning," in *2011 International Conference on Document Analysis and Recognition*, IEEE, 2011, pp. 440–445.

[18] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting," in *European Conference on Computer Vision (ECCV)*, 2016, pp. 512–528.

[19] W. Huang, Y. Qiao, and X. Tang, "Robust scene text detection with convolution neural network induced mser trees," in *European Conference on Computer Vision*, Springer, 2014, pp. 497–511.

[20] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," *International Journal of Computer Vision*, vol. 116, no. 1, pp. 1–20, 2016.

[21] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2315–2324.

[22] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai, "Multi-oriented text detection with fully convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4159–4167.

[23] X. Zhou et al., “East: an efficient and accurate scene text detector,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5551–5560.

[24] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, “Detecting text in natural image with connectionist text proposal network,” in *European Conference on Computer Vision*, Springer, 2016, pp. 56–72.

[25] S. Mahajan and R. Rani, “Text detection and localization in scene images: a broad review,” *Artificial Intelligence Review*, vol. 54, no. 6, pp. 4317–4377, 2021.

[26] S. Long, X. He, and C. Yao, “Scene text detection and recognition: The deep learning era,” *International Journal of Computer Vision*, vol. 129, no. 1, pp. 161–184, 2021.

[27] T. Khan, R. Sarkar, and A. F. Mollah, “Deep learning approaches to scene text detection: a comprehensive review,” *Artificial Intelligence Review*, vol. 54, no. 5, pp. 3239–3298, 2021.

[28] C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou, and Z. Cao, “Scene text detection via holistic, multi-channel prediction,” *arXiv preprint arXiv:1606.09002*, 2016.

[29] M. Bisht and R. Gupta, “Fine-Tuned Pre-Trained Model for Script Recognition,” *International Journal of Mathematical, Engineering and Management Sciences*, vol. 6, no. 5, p. 1297, 2021.

[30] M. Bisht and R. Gupta, “Handwritten Devanagari Word Detection and Localization using Morphological Image Processing,” in *2023 10th International Conference on Signal Processing and Integrated Networks (SPIN)*, IEEE, 2023, pp. 126–130.

[31] S. Ghosh and B. B. Chaudhuri, “Composite script identification and orientation detection for indian text images,” in *2011 International Conference on Document Analysis and Recognition*, IEEE, 2011, pp. 294–298.

[32] M. Verma, N. Sood, P. P. Roy, and B. Raman, “Script identification in natural scene images: a dataset and texture-feature based performance evaluation,” in *Proceedings of International Conference on Computer Vision and Image Processing*, Springer, 2017, pp. 309–319.

[33] L. Gomez and D. Karatzas, “A fine-grained approach to scene text script identification,” in *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, IEEE, 2016, pp. 192–197.

[34] B. Shi, C. Yao, C. Zhang, X. Guo, F. Huang, and X. Bai, “Automatic script identification in the wild,” in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, IEEE, 2015, pp. 531–535.

[35] B. Shi, X. Bai, and C. Yao, "Script identification in the wild via discriminative convolutional neural network," *Pattern Recognition*, vol. 52, pp. 448–458, 2016.

[36] D. Ghosh, T. Dube, and A. Shivaprasad, "Script recognition—a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, pp. 2142–2161, 2010.

[37] K. Ubul, G. Tursun, A. Aysa, D. Impedovo, G. Pirlo, and T. Yibulayin, "Script identification of multi-script documents: a survey," *IEEE Access*, vol. 5, pp. 6546–6559, 2017.

[38] N. Sharma, S. Chanda, U. Pal, and M. Blumenstein, "Word-wise script identification from video frames," in *2013 12th International Conference on Document Analysis and Recognition*, IEEE, 2013, pp. 867–871.

[39] N. Sharma, U. Pal, and M. Blumenstein, "A study on word-level multi-script identification from video frames," in *2014 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2014, pp. 1827–1833.

[40] Z. Li and J. Tang, "Unsupervised feature selection via nonnegative spectral analysis and redundancy control," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5343–5355, 2015.

[41] Z. Li, J. Liu, J. Tang, and H. Lu, "Robust structured subspace learning for data representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 10, pp. 2085–2098, 2015.

[42] Z. Li, J. Tang, and X. He, "Robust structured nonnegative matrix factorization for image representation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1947–1960, 2017.

[43] J. Mei, L. Dai, B. Shi, and X. Bai, "Scene text script identification with convolutional recurrent neural networks," in *2016 23rd International Conference on Pattern Recognition (ICPR)*, IEEE, 2016, pp. 4053–4058.

[44] A. K. Bhunia, A. Konwer, A. K. Bhunia, A. Bhowmick, P. P. Roy, and U. Pal, "Script identification in natural scene image and video frames using an attention based convolutional-LSTM network," *Pattern Recognition*, vol. 85, pp. 172–184, 2019.

[45] L. Lu, Y. Yi, F. Huang, K. Wang, and Q. Wang, "Integrating local CNN and global CNN for script identification in natural scene images," *IEEE Access*, vol. 7, pp. 52669–52679, 2019.

[46] M. Ma, Q.-F. Wang, S. Huang, S. Huang, Y. Goulermas, and K. Huang, "Residual Attention-Based Multi-Scale Script Identification in Scene Text Images," *Neurocomputing*, vol. 421, pp. 222–233, 2021.

[47] M. Tounsi, I. Moalla, F. Lebourgeois, and A. M. Alimi, “CNN based transfer learning for scene script identification,” in *International Conference on Neural Information Processing*, Springer, 2017, pp. 702–711.

[48] R. Bajaj, L. Dey, and S. Chaudhury, “Devnagari numeral recognition by combining decision of multiple connectionist classifiers,” *Sadhana*, vol. 27, no. 1, pp. 59–72, 2002.

[49] A. Elnagar and S. Harous, “Recognition of handwritten Hindu numerals using structural descriptors,” *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 15, no. 3, pp. 299–314, 2003.

[50] R. J. Ramteke and S. C. Mehrotra, “Feature extraction based on moment invariants for handwriting recognition,” in *2006 IEEE Conference on Cybernetics and Intelligent Systems*, IEEE, 2006, pp. 1–6.

[51] U. Garain, M. P. Chakraborty, and D. Dasgupta, “Recognition of handwritten indic script using clonal selection algorithm,” in *International Conference on Artificial Immune Systems*, Springer, 2006, pp. 256–266.

[52] N. Sharma, U. Pal, F. Kimura, and S. Pal, “Recognition of off-line handwritten devnagari characters using quadratic classifier,” in *Computer Vision, Graphics and Image Processing*, Springer, 2006, pp. 805–816.

[53] M. Hanmandlu and O. R. Murthy, “Fuzzy model based recognition of handwritten numerals,” *Pattern Recognition*, vol. 40, no. 6, pp. 1840–1854, 2007.

[54] M. Hanmandlu, A. V. Nath, A. C. Mishra, and V. K. Madasu, “Fuzzy model based recognition of handwritten hindi numerals using bacterial foraging,” in *6th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2007)*, IEEE, 2007, pp. 309–314.

[55] P. M. Patil and T. R. Sontakke, “Rotation, scale and translation invariant handwritten Devanagari numeral character recognition using general fuzzy neural network,” *Pattern Recognition*, vol. 40, no. 7, pp. 2110–2117, 2007.

[56] U. Pal, N. Sharma, T. Wakabayashi, and F. Kimura, “Handwritten numeral recognition of six popular Indian scripts,” in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, IEEE, 2007, pp. 749–753.

[57] U. Bhattacharya and B. B. Chaudhuri, “Handwritten numeral databases of Indian scripts and multistage recognition of mixed numerals,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 3, pp. 444–457, 2008.

[58] U. Pal, R. K. Roy, K. Roy, and F. Kimura, "Indian multi-script full pin-code string recognition for postal automation," in *2009 10th International Conference on Document Analysis and Recognition*, IEEE, 2009, pp. 456–460.

[59] S. Basu, N. Das, R. Sarkar, M. Kundu, M. Nasipuri, and D. K. Basu, "A novel framework for automatic sorting of postal documents with multiscript address blocks," *Pattern Recognition*, vol. 43, no. 10, pp. 3507–3521, 2010.

[60] G. G. Rajput and S. M. Mali, "Fourier descriptor based isolated Marathi handwritten numeral recognition," *International Journal of Computer Applications*, vol. 3, no. 4, pp. 9–13, 2010.

[61] S. Acharya, A. K. Pant, and P. K. Gyawali, "Deep learning based large scale handwritten Devanagari character recognition," in *2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, IEEE, 2015, pp. 1–6.

[62] D. Khanduja, N. Nain, and S. Panwar, "A hybrid feature extraction algorithm for devanagari script," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 15, no. 1, p. 2, 2016.

[63] M. Bisht and R. Gupta, "Multiclass recognition of offline handwritten Devanagari characters using CNN," *International Journal of Mathematical, Engineering and Management Sciences*, vol. 5, pp. 1429–1439, 2020.

[64] S. Arora, D. Bhatcharjee, M. Nasipuri, and L. Malik, "A two stage classification approach for handwritten Devnagari characters," in *International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)*, IEEE, 2007, pp. 399–403.

[65] M. Hanmandlu, O. R. Murthy, and V. K. Madasu, "Fuzzy Model based recognition of handwritten Hindi characters," in *9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications (DICTA 2007)*, IEEE, 2007, pp. 454–461.

[66] U. Pal, N. Sharma, T. Wakabayashi, and F. Kimura, "Off-line handwritten character recognition of devnagari script," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, IEEE, 2007, pp. 496–500.

[67] P. S. Deshpande, L. G. Malik, and S. Arora, "Fine Classification & Recognition of Hand Written Devnagari Characters with Regular

Expressions & Minimum Edit Distance Method.," *Journal of Computers*, vol. 3, no. 5, pp. 11–17, 2008.

[68] U. Pal, S. Chanda, T. Wakabayashi, and F. Kimura, "Accuracy improvement of Devnagari character recognition combining SVM and MQDF," in *11th International Conference on Frontiers in Handwriting Recognition*, Citeseer, 2008, pp. 367–372.

[69] S. Arora, D. Bhattacharjee, M. Nasipuri, D. K. Basu, M. Kundu, and L. Malik, "Study of different features on handwritten Devnagari character," in *2009 Second International Conference on Emerging Trends in Engineering & Technology*, IEEE, 2009, pp. 929–933.

[70] S. Kumar, "Performance comparison of features on Devanagari handprinted dataset," *International journal of recent trends in engineering*, vol. 1, no. 2, pp. 33–37, 2009.

[71] V. Mane and L. Ragha, "Handwritten character recognition using elastic matching and PCA," in *Proceedings of the International Conference on Advances in Computing, Communication and Control*, ACM, 2009, pp. 410–415.

[72] U. Pal, T. Wakabayashi, and F. Kimura, "Comparative study of Devnagari handwritten character recognition using different feature and classifiers," in *2009 10th International Conference on Document Analysis and Recognition*, IEEE, 2009, pp. 1111–1115.

[73] S. Arora, D. Bhattacharjee, M. Nasipuri, D. K. Basu, and M. Kundu, "Recognition of non-compound handwritten Devnagari characters using a combination of MLP and minimum edit distance," *International Journal of Computer Science and Security*, vol. 4, no. 1, pp. 107–120, 2010.

[74] S. Puri and S. P. Singh, "An efficient Devanagari character classification in printed and handwritten documents using SVM," *Procedia Computer Science*, vol. 152, pp. 111–121, 2019.

[75] S. P. Deore and A. Pravin, "Devanagari Handwritten Character Recognition using fine-tuned Deep Convolutional Neural Network on trivial dataset," *Sâdhanâ*, vol. 45, no. 1, p. 243, Sep. 2020, doi: 10.1007/s12046-020-01484-1.

[76] S. D. Pande et al., "Digitization of handwritten Devanagari text using CNN transfer learning–A better customer service support," *Neuroscience Informatics*, vol. 2, no. 3, p. 100016, 2022.

[77] M. Bisht and R. Gupta, "Offline handwritten Devanagari modified character recognition using convolutional neural network," *Sâdhanâ*, vol. 46, no. 1, pp. 1–4, 2021.

[78] M. Bisht and R. Gupta, "Conditional Generative Adversarial Network for Devanagari Handwritten Character Generation," in *2021 7th International Conference on Signal Processing and Communication (ICSC)*, IEEE, 2021, pp. 142–145.

[79] S. Basu, N. Das, R. Sarkar, M. Kundu, M. Nasipuri, and D. K. Basu, "A hierarchical approach to recognition of handwritten Bangla characters," *Pattern Recognition*, vol. 42, no. 7, pp. 1467–1484, Jul. 2009, doi: 10.1 016/j.patcog.2009.01.008.

[80] S. Malakar, P. Sharma, P. K. Singh, M. Das, R. Sarkar, and M. Nasipuri, "A Holistic Approach for Handwritten Hindi Word Recognition:," *International Journal of Computer Vision and Image Processing*, vol. 7, no. 1, pp. 59–78, Jan. 2017, doi: 10.4018/IJCVIP.2017010104.

[81] S. Bhowmik, S. Polley, M. G. Roushan, S. Malakar, R. Sarkar, and M. Nasipuri, "A holistic word recognition technique for handwritten Bangla words," *International Journal of Applied Pattern Recognition*, vol. 2, no. 2, pp. 142–159, 2015.

[82] H. Kaur and M. Kumar, "On the recognition of offline handwritten word using holistic approach and AdaBoost methodology," *Multimedia Tools and Applications*, vol. 80, no. 7, pp. 11155–11175, 2021.

[83] S. K. Parui and B. Shaw, "Offline handwritten devanagari word recognition: An hmm based approach," in *International Conference on Pattern Recognition and Machine Intelligence*, Springer, 2007, pp. 528–535.

[84] B. Shaw, S. K. Parui, and M. Shridhar, "Offline Handwritten Devanagari Word Recognition: A holistic approach based on directional chain code feature and HMM," in *2008 International Conference on Information Technology*, IEEE, 2008, pp. 203–208.

[85] B. Shaw, S. K. Parui, and M. Shridhar, "A segmentation based approach to offline handwritten Devanagari word recognition," in *2008 International Conference on Information Technology*, IEEE, 2008, pp. 256–257.

[86] B. Singh, A. Mittal, M. A. Ansari, and D. Ghosh, "Handwritten Devanagari word recognition: a curvelet transform based approach," *International Journal on Computer Science and Engineering*, vol. 3, no. 4, pp. 1658–1665, 2011.

[87] S. Ramachandrula, S. Jain, and H. Ravishankar, "Offline handwritten word recognition in Hindi," in *Proceeding of the workshop on Document Analysis and Recognition*, 2012, pp. 49–54.

[88] B. Shaw, U. Bhattacharya, and S. K. Parui, "Combination of features for efficient recognition of offline handwritten devanagari words," in *2014*

14th International Conference on Frontiers in Handwriting Recognition, IEEE, 2014, pp. 240–245.

[89] B. Shaw, U. Bhattacharya, and S. K. Parui, "Offline handwritten Devanagari word recognition: information fusion at feature and classifier levels," in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, IEEE, 2015, pp. 720–724.

[90] S. G. Oval and S. Shirawale, "Recognizing handwritten Devanagari words using recurrent neural network," in *Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014*, Springer, 2015, pp. 413–421.

[91] S. Kumar, "A study for handwritten Devanagari word recognition," in *2016 International Conference on Communication and Signal Processing (ICCSP)*, Melmaruvathur, Tamilnadu, India: IEEE, Apr. 2016, pp. 1009–1014. doi: 10.1109/ICCSP.2016.7754301.

[92] P. P. Roy, A. K. Bhunia, A. Das, P. Dey, and U. Pal, "HMM-based Indic handwritten word recognition using zone segmentation," *Pattern Recognition*, vol. 60, pp. 1057–1075, 2016.

[93] K. Dutta, P. Krishnan, M. Mathew, and C. V. Jawahar, "Offline Handwriting Recognition on Devanagari Using a New Benchmark Dataset," in *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, IEEE, 2018, pp. 25–30.

[94] A. Dwivedi, R. Saluja, and R. K. Sarvadevabhatla, "An OCR for classical Indic documents containing arbitrarily long words," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 560–561.

[95] R. K. Roy, H. Mukherjee, K. Roy, and U. Pal, "CNN based recognition of handwritten multilingual city names," *Multimedia Tools and Applications*, vol. 81, no. 8, pp. 11501–11517, 2022.

[96] M. Bisht and R. Gupta, "Offline Handwritten Devanagari Word Recognition Using CNN-RNN-CTC," *SN Computer Science*, vol. 4, no. 1, p. 88, 2022.

[97] B. V. Kasuba, D. Kudale, V. Subramanian, P. Chaudhuri, and G. Ramakrishnan, "PLATTER: A Page-Level Handwritten Text Recognition System for Indic Scripts," Feb. 10, 2025, *arXiv*: arXiv:2502.06172. doi: 10.48550/arXiv.2502.06172.

Biographies

Mamta Bisht is a passionate researcher and educator in Artificial Intelligence, Machine Learning, and Digital Communication. She earned her B.Tech. in Electronics and Communication Engineering from H.N.B. Garhwal University (A Central University, Srinagar, Uttarakhand) in 2011 and an M. Tech. in Digital Communication from B.T.K.I.T. Dwarahat (An Autonomous Institute of the Government of Uttarakhand) in 2015. She has pursued a Ph.D. in Electronics and Communication Engineering at Jaypee Institute of Information Technology, Noida, which she successfully completed in 2023. Her doctoral research, "Handwritten Text Recognition for Devanagari Script using Deep Learning Models," explores innovative solutions in pattern recognition and AI-driven linguistic processing. Her research contributions are featured in reputed indexed journals and conferences. She actively explores cutting-edge developments in AI, Image Processing, Pattern Recognition, Deep Learning, Signal Processing, and Communication. Currently, she serves as an Assistant Professor in the Department of Computer Science and Engineering (AI & ML) at Indraprastha Engineering College, Ghaziabad.

Richa Gupta is a Professor at Jaypee Institute of Information Technology (JIIT), Noida, with over 18 years of academic experience. Richa Gupta received her Ph.D. Degree from JIIT, Noida, India in the year 2013 and M.Tech in Information Systems from IIT, Kanpur, India in 2005. She obtained her B.Tech degree in ECE from KIET, Ghaziabad, India in 2003, with a special acknowledgement for being the University topper of B.R. Ambedkar University, Agra. She is a distinction holder in her entire academic career. Since August 2007, she has been teaching in the Department of Electronics and Communications Engineering, JIIT Noida. Since June 2024, she is working as a professor in the department. Prior to joining JIIT, she has worked with a USA based research company ‘ATC Labs, Noida’ for 2 years.