*"The aim of education is the knowledge not of facts but of values"*

*- William Ralph Inge*

# Chapter 2

# HISTORICAL PERSPECTIVE

## 2.1 OVERVIEW

The recognition of characters and words plays a significant role in pattern recognition research. Particularly, the task of recognizing handwritten characters and words is complex and challenging due to the diverse writing styles exhibited by individuals. The accuracy of such recognition systems greatly relies on the extraction and selection of appropriate features. Over the years, numerous researchers have proposed various feature extraction and classification methods for different scripts, including Devanagari. This chapter aims to provide a comprehensive review of the feature extraction and classification techniques that have been employed so far in both online and offline Handwritten Character Recognition (HCR) or Handwritten Word Recognition (HWR) of various scripts including Devanagari script. The Devanagari script holds substantial importance in the field of Optical Character Recognition (OCR) research. Within this chapter, the techniques employed, datasets utilized and achieved accuracies of existing methods proposed by different authors in OCR research are discussed. Furthermore, this chapter sheds light on the latest studies, identifies research gaps and highlights the challenges that still need to be addressed in this domain.

Section 2.2 of this chapter encompasses a comprehensive literature review that mainly covers various feature extraction and classification techniques. It provides an overview of the methods employed in previous research. In Section 2.3, a comparative analysis is presented, highlighting the feature wise findings obtained from the literature review. Moreover, this section identifies the existing research gaps that require further exploration. Lastly, Section 2.4 offers a detailed discussion and provides the valuable insights for the current research.

## 2.2 LITERATURE REVIEW

The literature review in this chapter has been categorized into two main groups: based on methods considered and based on recognition considered. The methods considered encompass feature extraction techniques, classification methods and the utilization of deep learning approaches. On the other hand, recognition considered include numeral/digit recognition, word/text recognition, isolated character recognition and script recognition. The literature review is organized into the following subsections for a comprehensive presentation.

### 2.2.1 Based on Methods Considered

The literature review based on methods considered include feature extraction techniques, classification methods and the utilization of deep learning approaches. Each of these aspects is discussed in detail in the subsequent subsections.

#### *2.2.1.1 Feature Extraction Methods*

This sub-section presents an overview of the feature extraction methods employed by different researchers in the specific research domain under investigation. Arica and Yarman-Vural, (2001) conducted feature extraction in handwritten Devanagari numeral/character recognition by calculating both statistical and structural features. Bajaj et al., (2002) focused on density, moment and descriptive component features for the same recognition task. Elnagar and Harous, (2003) employed end, branch and cross point features based on strokes and cavity information for recognizing handwritten Hindi numerals. Kaur, (2004) extracted features using Zernike moments and zoning techniques for the recognition of the Devanagari script. Kompalli et al., (2005 and 2006) employed Gradient, Structural and Concavity (GSC) features for recognizing machine-printed and multi-font Devanagari text. Ramteke and Mehrotra, (2006) extracted moment invariants as features, while Sharma et al., (2006) utilized directional chain code information of contour points for character recognition. Hanmandlu et al., (2007a and 2007b) proposed a box approach for recognizing handwritten numerals, which involved spatial division of numeral images into boxes. Furthermore, Pal et al., (2007a) employed chain code and gradient-based features for recognizing Devanagari numerals. In Pal et al., (2007b), utilized the arctangent of the gradient and Gaussian filter information as features for handwritten character recognition. More and Rege, (2008) recognized handwritten Devanagari numerals using simple geometric and Zernike moments. For handwritten Devanagari word recognition, Shaw et al., (2008a) utilized a feature vector based on the histogram of chain-code directions in image-strips. The image-strips were scanned from left to right using a sliding window approach.

Kumar, (2009) conducted a comparative analysis of different feature extraction methods, namely Kirsch directional edges, distance transforms chain code, gradient and directional distance distribution, using a dataset of handwritten Devanagari characters. Additionally, a novel feature was introduced in this study by quantizing the gradient direction into four directional levels. Each gradient map was further divided into $4 \times 4$ regions to facilitate the feature extraction process. Bhattacharya and Chaudhuri, (2008) employed contour representations of the four frequency components (high-high, high-low, low-high, and low-low) derived from wavelet-filtered images to extract high-level features for handwritten numeral recognition. Wakabayashi et al., (2009) proposed a feature extraction technique based on the Fisher-ratio (F-ratio) to achieve improved results for recognizing similar-shaped handwritten characters. Basu et al., (2010) conducted recognition and classification of handwritten digits using a Quad-Tree-based Longest Run (QTLR) feature. Rajput and Mali, (2010) utilized Fourier Descriptors (FD) as features for recognizing handwritten numerals. Arora et al., (2010a) recognized handwritten Devanagari compound characters by calculating shadow and chain code histogram features.

Aggarwal et al., (2012a) employed gradient representation as a feature extraction method for recognizing Devanagari characters. A dataset consisting of 7200 character samples was normalized to a size of $90 \times 90$ pixels. The experimental results using Support Vector Machines (SVM) demonstrated high performance, achieved a cross-validation accuracy of 94%. Pratap and Arya, (2012) provided a general overview of the Devanagari character recognition system. Pourmohammad et al., (2013) proposed an efficient character recognition system that utilizes Linear Discriminant Analysis (LDA) followed by a Bayesian discriminator function based on the Mahalanobis distance. The system employed Affine transformations on the training samples to enhance its robustness against scaling and rotation distortions. Dixit et al., (2014) proposed a unique recognition system that employed Wavelet features for classification and recognition of Devanagari characters. The system achieved a maximum accuracy of 70% over a dataset consisting of 2000 samples with 20 different letters. Singh and Maring, (2015) employed a combination of statistical and structured-based feature extraction techniques, including chain code, zone-based centroid, background directional distribution and distance profile features, for Devanagari Handwritten Character Recognition (HCR). They conducted experiments on a dataset comprising more than 20,000 samples with varying image sizes: $30 \times 30$, $40 \times 40$ and $50 \times 50$ pixels. The system achieved an overall accuracy of 97.61% using Support Vector Machine (SVM) for classification.

Ajmire et al., (2015) discussed statistical techniques suitable for feature extraction in handwritten character recognition. They explored various statistical approaches to extract discriminative features from handwritten characters. Tanuja et al., (2015) proposed a system for handwritten Hindi character recognition that employed canny edge detection, distance transformation and neural networks with back propagation algorithms. The system achieved an accuracy of 95.0% in recognizing handwritten Hindi characters. Ansari and Sutar, (2015) proposed an effective method for recognizing isolated Marathi handwritten words in the Devanagari script. Their approach involved extracting gradient, distance transforms, regional and geometric features from the images of handwritten words. These features were then used for classification using Feed-Forward Neural Network (FFNN) classifiers. The proposed method achieved an impressive overall recognition rate of 94.57%. However, some recognition errors were observed, particularly in cases of abnormal writing and ambiguity among similar-shaped words.

Wanchoo et al., (2016) discussed the challenges associated with automating the Indian postal system and presented a case study in this context. The paper highlighted the existing research literature that supports the development of postal automation solutions. Jangid and Srivastava, (2016) employed a novel masking technique along with the Fisher discrimination function and SVM classifier to extract features from the ISIDCHAR database, which contains standard Devanagari characters. Their approach significantly improved the recognition rate, achieving 96.58% of accuracy in similar character recognition. Kamble and Hegadi, (2016) proposed an approach for feature extraction in handwritten Marathi characters, a variant of Devanagari script. They utilized connected pixel-based features such as area, perimeter, eccentricity, orientation and Euler number. The authors compared the accuracy of their proposed methods and concluded that a modified SVM classifier outperformed the KNN classifier in terms of accuracy. In the work of Kumar et al., (2018b), the authors focused on recognizing 3D handwritten Devanagari words using the Bidirectional Long Short-Term Memory Neural Network (BLSTM-NN) classifier. They experimented with different features, including raw data, convex features, curvature features and writing direction features. Their recognition performance varied across the different feature sets, with the maximum accuracy achieved using 3D curvature features.

Jangid and Srivastava, (2018a) presented a novel system for recognizing handwritten Devanagari characters using Deep Convolutional Neural Networks (DCNN) and adaptive gradient methods. The DCNN architecture was employed to effectively extract and learn features from the input images, while the adaptive gradient methods enhanced the optimization process for improved performance and accuracy in character recognition. Kumar and Jindal, (2020) conducted a study focusing on the recognition of multi-lingual characters, including English, Hindi and Punjabi. They explored different features such as zoning, diagonal, horizontal peak extent based features, intersection points and open-end point based features. Additionally, they employed various classifiers, namely KNN, Linear-SVM, and MLP. The authors achieved recognition accuracies of 92.18%, 84.67% and 86.79% for English, Hindi and Punjabi characters, respectively.

Bhattacharyya et al., (2022) proposed a two-stage deep feature selection approach for the recognition of online handwritten Bangla and Devanagari basic characters. Authors obtained features using VGG-19 model and thereafter reduced using a two-stage feature selection approach i.e. the features are ranked using a filter method (first stage called ReliefF) and thereafter, the ranked features are optimized using gray wolf optimization (second stage). Their experimental results shows that the proposed approach reduces the feature dimension while simultaneously enhances the classification accuracy for both Bangla and Devanagari scripts. Raj et al., (2023) developed a framework for recognizing Tamil handwritten characters using locational as well as directional approaches. Authors explored zone, structural, locational and directional based features along with various classifiers. Their experiment showed significant character recognition results.

Many researchers proposed several classification methods that utilize features extracted for the character or script identification, as described below.

#### *2.2.1.2 Classification Methods*

The character or word recognition system encompasses a crucial decision-making step known as classification, which plays a vital role in determining the class membership of different characters or words for their recognition. In this subsection, the classification methods employed by various researchers in this specific domain are presented. Connell et al., (2000) achieved a recognition accuracy of 86.5% without any rejections by employing a combination of multiple classifiers that focus on either local on-line properties or global off-line properties of unconstrained Devanagari characters. Kaur, (2004) utilized a feed forward back propagation neural network, taking the feature vector as input, for the classification of handwritten Devanagari characters. Sharma et al., (2006) proposed a quadratic classifier-based method for this purpose. Pal et al., (2007b) introduced a modified quadratic classifier specifically designed for handwritten character recognition.

Arora et al., (2007) presented a two-stage classification approach for Devanagari character recognition. In the first stage, structural properties such as shirorekha (top line) and spine are extracted, while in the second stage, intersection features are utilized. The extracted features are then fed into a Feed-Forward Neural Network (FFNN) for classification. In their study, Hanmandlu, et al., (2007b) employed various features to classify Devanagari characters into three classes: end-bar, middle-bar, and characters without any bar, based on the presence of a vertical bar. This initial coarse classification step was performed before recognition and a modified exponential membership function was utilized to recognize handwritten characters by fitting it to the fuzzy sets resulting from the character features. The authors enhanced the learning process speed by implementing a reuse policy. Deshpande et al., (2008) introduced Regular Expressions (RE) as a valuable tool in handwritten Devanagari character recognition.

Their approach involved utilizing chain-code features to convert handwritten characters into an encoded string. To achieve improved accuracy in character recognition, Pal et al., (2008) combined two classifiers: Support Vector Machine (SVM) and Modified Quadratic Discriminant Function (MQDF). In their work, Shaw et al., (2008a) presented a continuous density Hidden Markov Model (HMM) approach for recognizing handwritten words. This method was applied to recognize entire handwritten words, providing a comprehensive solution for word recognition in Devanagari script.

Further, Shaw et al., (2008b) proposed a segmentation-based method for recognizing handwritten Devanagari words. The authors segmented word images into pseudo-characters based on the header line and subsequently recognized these pseudo-characters using Hidden Markov Models (HMM). More and Rege, (2008) introduced an Elastic Matching (EM) method based on Eigen Deformation (ED) for handwritten Devanagari character recognition. The EM method comprises two phases: training, for estimating ED and recognition. Pal et al., (2009a) proposed a dynamic programming-based method for recognizing pin code strings. In a comparative study conducted by Pal et al., (2009b), various classifiers were employed for Devanagari HCR, including Compound MQDF (CMQDF), Compound Projection Distance (CPD), Euclidean Distance, KNN, Linear Discriminant Function (LDF), Mirror Image Learning (MIL), Modified Projection Distance (MPD), MQDF, nearest neighbor, Projection Distance (PD), Sub-space method and Support Vector Machine (SVM). The study concluded that the Mirror Image Learning (MIL) classifier yielded the best results, while ED exhibited the lowest performance among the aforementioned classifiers.

A divide-and-conquer approach was implemented for Devanagari HCR by Agrawal et al., (2009). Hanmandlu et al., (2009) classified the top modifiers of Devanagari script as either one-touching-point or two-touching-point modifiers. Classification was further conducted by examining the core strip of the word. Arora et al., (2010a) classified Devanagari non-compound handwritten characters using two Multi-Layer Perceptrons (MLPs) and a Minimum Edit Distance (MED) method. The first phase involved using two MLPs to classify distinctly shaped characters, while in the second phase, similarly shaped characters were classified using the MED method. Shelke and Apte, (2010) proposed a Devanagari text recognition method using multistage feature extraction and classification techniques. The initial step involved extracting structural features, followed by Radon and Euclidean distance transforms. Two separate feed-forward back propagation neural networks were then applied to these features. The hybrid classifier, combining input from the two neural network classifiers and a template matching classifier, produced the final output based on a maximum voting rule. This method significantly improved recognition accuracy, achieving a recognition rate of 95.40%. Kubatur et al., (2012) achieved a recognition rate of up to 97.2% by employing a neural network-based framework for HCR. Kale et al., (2013) achieved an overall recognition rate of 98.25% and 98.36% for basic and compound characters, respectively, using Legendre moments as feature descriptors and Artificial Neural Networks (ANN) as classifiers. A novel part-based method was proposed by Narang et al., (2013) for recognizing Devanagari characters by identifying 40 basic classes. The problem of recognizing a very large class of instances has been addressed by training models to classify individual instances within the given test samples. This approach demonstrates competitive performance compared to state-of-the-art features and classifiers for the DSIW2K dataset.

Jangid and Srivastava, (2014) explored the Gradient Local Auto-Correlation (GLAC) algorithm for Handwritten Character Recognition (HCR) of Devanagari script, utilizing two databases: ISIDCHAR and V2DMDCHAR. The proposed method achieved the best results using the Support Vector Machine (SVM) classifier, with recognition accuracies of 93.21% for ISIDCHAR and 95.21% for V2DMDCHAR. Dongre and Mankar, (2015) employed a Multi-Layer Perceptron Neural Network (MLP-NN) as a classifier for recognizing Devanagari numerals and characters. They achieved recognition accuracies of 93.17% (using 40 hidden neurons) and 82.7% (using 60 hidden neurons) for numerals and characters, respectively, based on the structural and geometric features extracted. Ghosh and Roy, (2015a) focused on online HCR of Bengali and Devanagari scripts by extracting structural and directional features individually in each local zone. These features were concatenated and fed to an SVM classifier, resulting in recognition accuracies of 87.48% for Bengali and 84.10% for Devanagari, using 4,900 and 5,000 test samples, respectively. In another work by Ghosh and Roy, (2015b), two zone-based feature extraction methods, namely Zone-wise Structural and Directional (ZSD) and Zone-wise Slopes of Dominant Points (ZSDP), were presented for online HCR of Bengali and Devanagari scripts. The SVM classifier was employed for stroke recognition, and character recognition was performed based on stroke combinations with training data. The recognition accuracies achieved with ZSD were 87.48% for Bengali (with 9,800 test data) and 85.10% for Devanagari (with 10,000 test data), while with ZSDP, the accuracies improved to 92.48% for Bengali and 90.63% for Devanagari, respectively.

Pagare and Verma, (2015) implemented a dynamic model based on a Hopfield neural network for auto-associative recognition of Devanagari characters and numerals. Shelke and Apte, (2015) proposed a novel multi-stage classification approach for recognizing unconstrained handwritten Devanagari characters. This approach involves a fuzzy inference system in the first step and utilizes structural parameters in the second step. The method achieved a recognition accuracy of 96.95%. Shelke and Apte, (2016) presented techniques to optimize recognition accuracy at different stages, including pre-classification, feature extraction and recognition. Various structural features were used for pre-classification, followed by optimized feature extraction methods. Finally, a neural network was employed for recognition. The performances of different neural networks were analyzed in this study. Kumar et al., (2018b) recognized 3D handwritten Latin and Devanagari words using multiple Bidirectional Long-Short Term Memory Neural Network (BLSTM-NN) classifiers and the Recognizer Output Voting Error Reduction (ROVER) framework. Their lexicon-free approach achieved accuracies of 72.25% for Latin and 71.86% for Devanagari. Jangid and Srivastava, (2018b) proposed a method for the recognition of handwritten Devanagari characters using gradient-based features and an SVM classifier. They achieved 96.58% recognition accuracy on ISIDCHAR dataset in their work.

Gupta and Bag, (2019), in their work, achieved Hindi character recognition accuracies of 95.10% using random forest, 95.57% using SVM, 96.09% using MLP and 94.71% using Convolutional Neural Network (CNN) classifiers. Narang et al., (2019b) presented a paper to recognize the Devanagari ancient manuscripts using AdaBoost and Bagging techniques. Authors achieved maximum 91.70% of recognition accuracy using adaptive boosting with RBF (Radial Basis Function)-SVM. Narang et al., (2020) achieved 91.39% recognition accuracy based on the tenfold cross-validation technique and poly-SVM classifier for the recognition of Devanagari ancient characters. In the field of statistics and machine learning (Sethi and Kaushik, 2020), classification algorithms, including Naive Bayes Classifier, Nearest Neighbour, Logistic Regression, Decision Trees, Random Forest, Neural Network, and KNN Classification, are commonly used to analyze the training database and perform classification on the testing/target database. Devi et al., (2021) investigated the performance of different machine learning algorithms, including Random Forest (RF), Logistic Regression (LR), Support Vector Machine (SVM) and K-Nearest Neighbor (KNN), for the recognition of handwritten characters. Through their experiments, the authors found that the KNN-based system achieved the highest recognition accuracy of 98%.

Singh et al., (2021) investigated stroke classification using a Recurrent Neural Network (RNN) classifier for the recognition of Gurmukhi words. The authors reported an accuracy rate of 98.67% on their collected dataset of 52,570 Gurmukhi words. Ramteke et al., (2022) proposed an OCR framework for handwritten Marathi document classification and recognition. They utilized a Weighted One-Against-Rest Support Vector Machine (WOAR-SVM) classifier to handle the large number of features extracted from the preprocessed images. Experimental results showed that the combination of various features (statistical, global transformation, geometrical and topological features) with the WOAR-SVM classifier achieved an accuracy of 95.14%.

### *2.2.1.3 Deep Learning Methods*

The deep learning methods have led to advancements in various applications, including Optical Character Recognition (OCR), document analysis and natural language processing due to its ability to achieve state-of-the-art performance, surpassing traditional machine learning techniques. Jangid and Srivastava, (2018a) developed a recognition technique for handwritten Devanagari characters using a layer-wise approach of Deep CNN, achieving higher recognition accuracy and faster convergence compared to shallow handcrafted feature-based methods and standard Deep CNN. Deore and Pravin, (2020) created a dataset consisting of 5,800 isolated images representing 58 unique character classes, which included 12 vowels, 36 consonants and 10 numerals. The authors employed a two-stage VGG16 deep learning model for the recognition of Devanagari handwritten characters. The first model achieved a testing accuracy of 94.84% with a training loss of 0.18, while the second model achieved a testing accuracy of 96.55% with a training loss of 0.12. Narang et al., (2021) proposed a deep learning model based on Convolutional Neural Networks (CNNs) for recognizing ancient texts written in the Devanagari script. They conducted experiments using a dataset containing 5,484 characters and achieved a recognition accuracy of 93.73% by employing CNN as a feature extractor. Mishra et al., (2021) classified handwritten Devanagari characters using a ResNet architecture having 85 convolution layers. They achieved 99.72% of accuracy on a publicly available dataset (named as Devanagari Handwritten Character Dataset (DHCD)) with 92,000 images (46 classes). Bisht and Gupta, (2021) proposed two models based on single CNN architecture and double-CNN architecture for the recognition of offline handwritten modified Devanagari characters Through their experimentation, the authors observed that double-CNN architecture performs better than single CNN architecture and uses a reduced number of output classes as compared with the actual existing classes of the same. Authors concluded that their proposed CNN architecture yields better competitive results as compared with the traditional feature extraction (HoG) and classification (SVM) techniques.

Pande et al., (2022) configured a Convolutional Neural Network (CNN) with appropriate strategies to effectively recognize handwritten Devanagari writing. Their 12-layer CNN approach, including a Dropout Layer, achieved an accuracy of 99.13% for 46 classes of Devanagari characters. Deore, (2022) proposed a technique for the recognition of handwritten Devanagari words using scan profile and sliding window approaches. The author employed ResNet as a classifier and achieved an accuracy of 86%. Ansari et al., (2022) presented an approach for the recognition of handwritten Devanagari characters. The authors adopted a two-step process, where they first extracted deep features using a Deep Convolutional Neural Network (Deep CNN) and thereafter, Support Vector Machine (SVM) classification approach was employed to classify the characters based on the extracted features. The experimental results showcased good performance, with a recognition accuracy of 99.41% on their real-word dataset.

### 2.2.2 Based on Recognition Considered

The field of Optical Character Recognition (OCR) for Indian languages/scripts has consistently demanded extensive research efforts, despite encountering numerous challenges and limited commercial opportunities (Jayadevan et al., 2012). Over the past few years, several researchers have developed various methods for recognizing numeral/digit, words/text, isolated characters and scripts. This section provides a brief overview of these methods.

#### *2.2.2.1 Numeral/Digit Recognition*

Hanmandlu and Murthy, (2007a) proposed a handwritten recognition system for Hindi and English numerals based on a fuzzy model. The authors achieved overall recognition rates of 95% and 98.4% for Hindi and English numerals, respectively. Bhattacharya and Chaudhuri, (2008) developed two databases containing handwritten numerals from the Devanagari and Bangla scripts, with sample sizes of 22,556 and 23,392, respectively. They also introduced a multistage cascaded recognition scheme that utilized wavelet-based multiresolution representations and Multi-Layer Perceptron (MLP) classifiers to recognize mixed handwritten numerals from Devanagari, Bangla and English scripts. Niu and Suen, (2012) proposed a hybrid model for recognizing handwritten digits, combining a Convolutional Neural Network (CNN) for trainable feature extraction and a Support Vector Machine (SVM) as the recognizer. They used the MNIST digit database for feature extraction and achieved recognition rates of 99.81% (without rejection) and 94.40% (with 5.60% rejection). Aggarwal et al. (2012b) developed an isolated handwritten Devanagari numerals recognition system based on gradient features and an SVM classifier. They achieved an accuracy of 99.60% using a standard dataset provided by the Indian Statistical Institute (ISI) Kolkata.

Arya et al., (2015) presented an offline Devanagari handwritten numeral recognition system using Gabor filters for feature extraction and employed nearest neighbor and SVM classifiers for classification. The authors tested three filter sizes ($7 \times 7$, $19 \times 19$, and $31 \times 31$) to determine the optimal size and achieved a maximum recognition accuracy of 98.06%. Dongre and Mankar, (2015) introduced a system for recognizing Devanagari numerals and characters using structural and geometric features. They employed an MLP-NN as the classifier and conducted experiments on a dataset of 3000 handwritten samples of Devanagari numerals, achieving a recognition accuracy of 93.17% using 40 hidden neurons. Kumar et al., (2019a) conducted a comprehensive survey on character and numeral recognition of non-Indic and Indic scripts, highlighting the major challenges and issues associated with character/numeral recognition. Ahlawat et al., (2020) proposed a Convolutional Neural Network (CNN) architecture for the recognition of handwritten digits. They explored the combination of various learning parameters while designing a CNN architecture. Authors achieved 99.87% of recognition accuracy using MNIST dataset, which consists of handwritten digits. They claimed that obtained accuracy surpasses that of ensemble architectures and offers operational complexity as well cost.

Prashanth et al., (2020) proposed an approach for the classification of handwritten numerals written using in the Devanagari script using artificial neural networks. They developed a new dataset specifically for handwritten Devanagari numerals which consists of 4,282 numerals collected from individuals of various age groups. Authors achieved an accuracy of over 95% and results has been compared with existing available datasets. Haghighi and Omranpour, (2021) proposed a new model for recognizing handwritten digits (Persian/Arabic) based on a stacking ensemble classifier. This classifier is based on Convolutional Neural Network (CNN) and the Bidirectional Long-Short Term Memory (BLSTM). Authors achieved 99.98% (training) and 99.39% (testing) of recognition accuracies for a Persian/Arabic dataset of comprising 102,352 data.

Prashanth et al., (2022) developed an approach for the recognition of handwritten Devanagari characters (numerals and vowels). Initially, they gathered a corpus of 38,750 images and thereafter, experiments are conducted on the same using three different CNN architectures: CNN, Modified LeNet CNN (MLCNN) and AlexNet CNN (ACNN). They have achieved a recognition rate of 96% and 94% on training and unseen data using CNN, respectively. Whereas, MLCNN attained 99% (training) and 94% (unseen data) of accuracy rate with less computational cost. Moreover, ACNN reached 99% (training) and 98% (unseen data) of recognition rate on unseen data.

#### *2.2.2.2 Word/Text Recognition*

Parui and Shaw, (2007) proposed a Hidden Markov Model (HMM) for recognizing handwritten Devanagari words using stroke-based features. Unlike traditional HMMs, the states of this model are not predefined but automatically determined based on a database of handwritten word images. Each word is treated as a sequence of stroke primitives, which serve as the states in the HMM and are identified using mixture distributions. The proposed classification scheme was evaluated on an in-house database of handwritten Devanagari words. The training dataset achieved a classification accuracy of 87.71%, while the testing dataset achieved an accuracy of 82.89%. Shaw et al., (2008a) proposed an approach for recognizing offline handwritten Devanagari words using a HMM. The feature vector was extracted by analyzing the histogram of chain-code directions in image-strips scanned from left to right. A continuous density HMM was employed for word recognition, where the states of the HMM were automatically determined based on a database of handwritten word images. Each word was represented as a sequence of image frame primitives. They conducted experiments on a dataset of 22,500 training words and 17,200 test words, achieved an accuracy of 80.2%. In a subsequent work (Shaw et al., 2008b), they utilized stroke-based features and an HMM classifier on the same dataset, achieved an accuracy of 84.31%.

Shaw and Parui, (2010) proposed a two stage recognition scheme for the recognition of offline handwritten Devanagari words. They considered a corpus of 13,000 words for experimental purpose keeping 7,000 words as training samples, 3,000 words as testing samples and other 3,000 words for validation. Authors achieved 85.57% (testing) and 91.25% (training) accuracies using Stroke based (stage-1) and Wavelet (stage-2) based feature extraction techniques and HMM classification. Singh et al., (2011) proposed an offline handwritten Devanagari word recognition system that addresses the challenges posed by the large variety of symbols and their visual similarity. The Curvelet Transform was employed to extract features capable of distinguishing similar appearing words. To handle the resulting high-dimensional feature space, Principal Component Analysis (PCA) was applied. The SVM and KNN classifiers were used. Experimental results demonstrated that the Curvelet features combined with the KNN classifier achieved the highest accuracy of 93.21% on a dataset of 28,500 handwritten Devanagari words.

Patel and Desai, (2011) proposed a technique based on zone identification for recognition of handwritten Gujarati words. They identified and extracted various zones namely upper, middle and lower zones based on distance transform. These zones represent upper modifiers, base character itself and lower modifiers. Authors achieved accuracies of 75.2% for both upper and middle zones, whereas they gained 83.6% of accuracy for lower zone of the word using Euclidean distance transform. Further, they also suggested that accuracy may be improved by applying preprocessing step namely slant correction. Ramachandrula et al., (2012) developed a system for the Recognition of offline handwritten Hindi words. To facilitate the training and testing of the HWR, they created a database consisting of handwritten Hindi words and characters from 100 writers. Their HWR system utilizes a two-pass Dynamic Programming algorithm, where the test word is matched against each word in the lexicon by initially segmenting the test word image into probable characters. They extract Directional Element Features (DEF) from each character image segment and model them statistically. Their system achieved word recognition accuracies ranging from 91.23% to 79.94% on vocabulary sizes ranging from 10 to 30 words.

Shaw et al., (2014) proposed an efficient approach for the recognition of handwritten Devanagari words. They considered combination of skeleton and contour based features along with SVM classification and obtained 79.01% of recognition accuracy on the corpus of 39,700 words. Bhowmik (2014b) proposed a technique for handwritten Bangla word recognition based on a holistic approach. They extracted elliptical features from the entire word to construct a feature space and achieved 85.88% recognition accuracy considering the 3-fold cross-validation method on 680 training and 340 testing samples. Bhowmik et al., (2015) proposed a recognition method for handwritten Bangla words based on concentric rectangles and convex hull-based features. Authors collected 2,754 handwritten Bangla word samples from different writers as databases for their work. They used a neural network-based classifier (3-fold cross-validation) and achieved 84.74% recognition accuracy.

Kadhm and Hassan, (2015) proposed an architecture for the recognition of handwritten words written in Arabic (AHDB database). They used statistical features (connected components and zoning features) and SVM polynomial kernel as classifier for word level recognition. Authors achieved recognition accuracy of 96.317% on AHDB database considering 70% as training and 30% as testing samples. Shaw et al., (2015) presented an offline recognition technique for handwritten Devanagari words using Directional Distance Distribution (DDD) and Gradient-Structural Concavity (GSC), focusing on 100 Indian town names and significantly improving recognition accuracy.

Kumar, (2016) proposed a segmentation-based approach for recognizing isolated hand-printed Devanagari words, utilizing a database of more than 3,500 words and classification was performed using a Multi-Layer Perceptron (MLP). Khemiri et al., (2016) proposed a system for recognition of Arabic words using various structural features. Authors explored Bayesian networks using a corpus of Tunisian city names (IFN-ENIT). Authors obtained 90.02% of recognition rate using Vertical/Horizontal-HMM (VH-HMM). Adak et al., (2016) presented a recognition system for handwritten word recognition written in Bengali. Authors explored CNN (Convolutional Neural Network) and recurrent model for the same. They analyzed the performance of their system on three different types of datasets. Authors claimed that combination of CNN and recurrent model give better performance.

Paneri et al., (2017) proposed a recognition technique for Gujarati handwritten words based on Histogram of Oriented Gradients (HOG) features. Authors explored SVM and KNN (K-Nearest Neighbor) classification techniques and obtained maximum accuracy of 85.87% and 76.87%, respectively. Bhunia et al., (2018) proposed a cross-language platform for handwritten word recognition and spotting in low-resource scripts. The framework enables training on a large dataset from one script and testing on another script with limited training data. They focused on three Indic scripts: Bangla, Devanagari, and Gurumukhi. By employing zone-wise character mapping and an entropy-based script similarity score, the framework achieved feasible cross-language transcription. The approach demonstrated promising results in recognizing and spotting text in target scripts where sufficient training data was not available.

In their study, Ghosh et al., (2019) devised a segmentation-free method for the recognition of handwritten Bangla word images. They employed a wrapper-filter approach based on the Memetic Algorithm (MA) to enhance the classification and dimensionality reduction feature vector. The authors utilized grid-based gradient and Statistical and Contour based Features (SCF), along with an MLP classifier, achieving a maximum recognition accuracy of 93% for a dataset of 50-city names (7,500 words) written in Bangla script. Bhowmik et al., (2019) developed a system for the recognition of handwritten Bangla words using holistic based approach so as to minimize the problems associated with character-level segmentation. Authors explored combination of elliptical, tetragonal and vertical-pixel density histogram-based features along with

MLP and SVM classifiers using a corpus of 18,000 words. They obtained 83.64% of recognition accuracy. Malakar et al., (2020a) developed a Hierarchical Feature Selection (HFS) model using a genetic algorithm to optimize the extracted local and global features of handwritten word images. They utilized a database comprising 12,000 samples of Bangla words to construct feature descriptors based on the shape/texture of the handwritten text images. Through feature dimension reduction of up to 28%, they improved the recognition performance of handwritten word recognition (HWR) by 1.28%. The authors claimed a recognition accuracy of 95.30% using gradient-based and elliptical features, along with MLP classification.

Kaur and Kumar, (2021a) developed a holistic approach based on eXtreme Gradient Boosting (XGBoost) for offline Handwritten Word Recognition (HWR) of Gurumukhi words. They extracted zoning, diagonal, intersection & open-end points, and peak extent features from a database of 40,000 samples of Gurumukhi words and employed the XGBoost approach for classification. The proposed system achieved an accuracy of 91.66%. Kaur and Kumar, (2021b) explored adopted majority voting and boosting algorithms to develop handwritten Gurumukhi words recognition system. They used KNN, SVM and random forest classification approaches for their work. Authors achieved recognition accuracy of 88.78% on the in-house database of 1,00,000 images of handwritten Gurumukhi words.

Sharma et al., (2022) used CNN model and various optimizing methods such as Adam and Stochastic Gradient Descent (SGD) for the recognition of Gurumukhi words (city names). They achieved recognition accuracy of 99.13% on the database of 4,000 handwritten Gurumukhi word images for their proposed system. Korichi et al., (2022) proposed a Generic Feature Independent Pyramid Multilevel Model (GFIPML) for recognition of Arabic words. Authors claimed that GFIPML gains the benefits of both Multi-Level (ML) and Pyramid Multi-Level (PML) features extraction schemes. Experiments were conducted using AHDB dataset and achieved 96.5% of recognition accuracy along with Linear Discriminant Analyses (LDA).

Table 2.1 showcases the recognition outcomes achieved by various researchers in the domain of Devanagari and other scripts for word/text recognition, employing diverse features and classifiers.

**Table 2.1:** Recognition results of word/text

| Authors | Script or Language | Dataset (Words) | Approach | | Recognition Accuracy (%) |
|---|---|---|---|---|---|
| | | | Feature Extraction | Classification | |
| Parui and Shaw, (2007) | Devanagri *(Offline)* | 7,000 *(Training)* 3,000 *(Testing)* | Stroke-based | HMM | 87.71% (*Training)* 82.89% *(Testing)* |
| Shaw et al., (2008a) | Devanagri *(Offline)* | 22,500 *(Training)* 17,200 *(Testing)* | Directional Chain Code-based | HMM | 80.20% |
| Shaw et al., (2008b) | Devanagri *(Offline)* | 22,500 *(Training)* 17,200 *(Testing)* | Stroke-based | HMM | 84.31% |
| Shaw and Parui, (2010) | Devanagri *(Offline)* | 7,000 *(Training)* 3,000 *(Testing)* 3,000 *(Validation)* | Stroke-based *(Stage-1);* Wavelet-based *(Stage-2)* | HMM *(Stage-1);* Modified Byes *(Satge-2)* | 85.57% *(Testing);* 91.25% *(Training)* *(Stage-2)* |
| B. Singh et al., (2011) | Devanagri *(Offline)* | 28,500 | Curvelet Transform-based | (a) SVM and (b) KNN | (a) 85.60% (b) 93.21% |
| Patel and Desai, (2011) | Gujarati *(Offline)* | 250 | Zone Identification -based | Euclidean Distance Transform | (a) 75.20% *(Upper zone);* (b) 75.20% *(Middle zone);* (c) 83.60% *(Lower zone)* |
| Ramachand rula et al., (2012) | Hindi *(Offline)* | 39,600 | Directional Element-based | Dynamic Programming | 79.94% *(30 Vocabulary Words);* 91.23% *(10 Vacbulary Words)* |
| Shaw et al., (2014) | Devanagri *(Offline)* | 22,500 *(Training)* 17,200 *(Testing)* | Combination of Skeleton and Contour-based | SVM | 79.01% |
| Bhowmik, Roushan, et al., (2014) | Bangla *(Offline)* | 680 *(Training)* 340 *(Testing)* | Histograms of Oriented Gradients (HOG)-based | MLP | 87.35% |
| Bhowmik et al., (2015) | Bangla *(Offline)* | 1,836 *(Training)* 918 *(Testing)* | Concentric Rectangles and Convex Hull-based | MLP | 84.74% |
| Kadhm and Hassan, (2015) | Arabic *(Offline)* | 2,044 *(Training)* 896 *(Testing)* *Arabic Handwriting Database (AHDB)* | Integration of using Multi Scale Features *(Connected Components, Zoning, DCT, HOG-based)* | SVM | 96.31% |

| | | | | | |
|---|---|---|---|---|---|
| Shaw et al., (2015) | Devanagri *(Offline)* | 22,500 *(Training)* 17,200 *(Testing)* | DDD and GSC-based | Multiclass SVM | 88.75% |
| Kumar, (2016) | Devanagari *(Isolated Hand Printed)* | More than 3500 | Chain Codes, Cumulative Histograms, Gradient, Neighbor Pixel Weight-based | MLP | 80.80% *(for Two Character Words)* 72.00% *(for Six Character Words)* |
| Khemiri et al., (2016) | Arabic *(Offline)* | 5,254 *(Training)* 2,627 *(Testing)* *(IFN/ENIT: Tunisian City Names Dataset)* | Structural | Bayesian Networks *(Naive Bayes, Tree Augmented Naive Bayes Network, Horizontal and Vertical HMM and Dynamic Bayesian Network)* | 90.02% *(Horizontal and Vertical HMM)* |
| Adak et al., (2016) | Bengali *(Offline)* | (a) 17,091 *(Public)* (b) 1,07,550 *(In-House)* (c) 5,223 *(Unconstrained)* | CNN-derived Features | RNN | (a) 85.42% (b) 86.96% (c) 70.67% |
| Paneri et al., (2017) | Gujarati *(Offline)* | 2,700 | Histogram of Oriented Gradients (HoG)-based | (a) SVM and (b) KNN | (a) 85.87% (b) 76.87% |
| Bhunia et al., (2018) | Bangla, Devanagari and Gurumukhi *(Offline)* | 3,856; 3,589 and 3,142 | PHOG-based | HMM *(for Middle-Zone);* SVM *(for Upper/ Lower Zone )* | Above 60.00% |
| Ghosh et al., (2019) | Bangla *(Offline)* | 6,000 *(Training)* 1,500 *(Testing)* | Gradient and Modified SCF; MA-based Wrapper Filter Selection | MLP | 93.00% |
| Bhowmik et al., (2019) | Bengali *(Offline)* | 14,400 *(Training)* 3,600 *(Testing)* | Combination of various Elliptical, Tetragonal and Vertical pixel density histogram-based features | (a) MLP (b) SVM | (a) 81.72% (b) 83.64% |

| | | | | | |
|---|---|---|---|---|---|
| Malakar et al., (2020a) | Bangla *(Offline)* | 12,000 | Gradient and Elliptical-based | MLP | 95.30% |
| Kaur and Kumar, (2021a) | Gurumukhi *(Offline)* | 40,000 | (a) Zoning-based (b) Diagonal-based (c) Intersection & Open-End Points-based (d) Peak Extent-based | XGBoost | (a) 91.66% (b) 91.30% (c) 88.37% (d) 86.27% |
| Kaur and Kumar, (2021b) | Gurumukhi *(Offline)* | 80,000 *(Training)* 20,000 *(Testing)* | Zoning, Diagonal and Intersection & Open-End Point-based | KNN, RBF-SVM, Random Forest, Majority Voting, AdaBoost | 88.78% *(AdaBoost)* |
| Sharma et al., (2022) | Gurumukhi *(Offline)* | 3,200 *(Training)* 800 *(Testing)* | Convolutional Neural Network | (a) Adam Optimizer (b) Stochastic Gradient Descent (SGD) Optimizer | (a) 99.13% (b) 94.18% |
| Korichi et al., (2022) | Arabic *(Offline)* | 6,615 *Arabic Handwriting Database (AHDB)* | Generic Feature-Independent Pyramid Multi-Level (GFIPML) | Linear Discriminant Analysis (LDA) | 96.50% |

### *2.2.2.3 Isolated Character Recognition*

Hanmandlu et al., (2007b) investigated the application of fuzzy models for handwritten Hindi character recognition. Their study incorporated a reuse policy that leverages past errors to enhance reinforcement learning and expedite the convergence of the learning process. By combining this policy with reinforcement learning, a substantial 25-fold improvement in training was observed. The experimentation was conducted on a database comprising 4,750 samples, resulting in better overall recognition rate of 90.65%. The findings of their research highlight the effectiveness of fuzzy models and the significant impact of the reuse policy in improving the efficiency and accuracy of handwritten Hindi character recognition. Pal et al., (2008) proposed a recognition system for offline handwritten Devanagari characters, employing a combination of Modified Quadratic Discriminant Function (MQDF) and Support Vector Machine

(SVM) classifiers based on gradient and curvature features. Their system achieved an accuracy of 95.13% on a dataset of 36,172 samples. Agrawal et al., (2009) presented a system for the classification of off-line handwritten Hindi characters using a similarity measure. The paper proposed a novel method to identify the header line by analyzing the end points and pixel positions in the top half of the character image, even in the presence of slant. Thereafter, the characters were subjected to coarse classification. The authors also introduced an algorithm to detect the presence and position of a vertical bar in handwritten Hindi characters. Through simulation studies, a classification rate of 97.25% was achieved.

Kubatur et al., (2012) proposed a Neural Network (NN)-based framework for the classification of online Devanagari characters into the 46 characters of the alphabet set. Authors introduced three key contributions: firstly, the feature extraction utilizing the Discrete Cosine Transform (DCT); secondly, the mode of character input was through a computer mouse and lastly, the researchers constructed a database. The proposed framework was evaluated on a database of 2,760 characters, achieving recognition rates of up to 97.2%. Yadav et al., (2013) conducted a study focusing on improving the recognition rate of printed Hindi texts through the utilization histogram of projection based features. It was specifically designed to extract robust features, even in the presence of character distortions or variations. To develop the classification model, a back-propagation neural network with two hidden layers was employed. They achieved recognition rates of 98.5% at the individual letter level and 90% at the paragraph level.

Kale et al., (2013) proposed a recognition system for handwritten Devanagari compound characters using Legendre moment feature descriptors. The system achieved high recognition rates of 98.25% for 3,750 basic Devanagari characters and 98.36% for 11,250 compound characters. The process involved normalizing the input image, dividing it into zones and extracting structural and statistical features from each zone. An Artificial Neural Network (ANN) was used for classification. Dixit et al., (2014) proposed an approach for the recognition of handwritten Devanagari characters. Their experimentation involved using 20 handwritten characters from 100 individuals, resulting in a dataset of 2,000 characters. Wavelet transform was applied to each individual character to obtain decomposed images. Statistical parameters were computed from the decomposition to form feature vectors. These feature vectors served as input to a Back Propagation Neural Network (BPNN) for classification into one of the 20-classes and subsequent recognition. They obtained 70% of accuracy over a dataset of 2,000 characters (samples) in recognizing handwritten Devanagari characters. Jangid and Srivastava, (2014) focused on utilizing the object detection algorithm GLAC (Gradient Local Auto-Correlation) for handwritten character recognition. In this study, GLAC in conjunction with the SVM classifier, was applied to two handwritten Devanagari databases, namely ISIDCHAR and V2DMDCHAR. They achieved 93.21% and 95.21% of recognition accuracies on the ISIDCHAR and V2DMDCHAR databases, respectively. These results provide evidence for the effectiveness of the GLAC algorithm in addressing the character recognition problem.

Acharya et al., (2015) introduced the Devanagari Handwritten Character Dataset (DHCD) which comprises 92, 000 images representing 46 different classes of characters. They also proposed a deep learning architecture for character recognition. Departing from the conventional approach to character recognition using Deep CNNs, the authors focused on utilizing Dropout and dataset increment techniques to enhance test accuracy. By incorporating these techniques into their Deep CNN architecture, authors achieved a notable increase in test accuracy, with the proposed model achieving the highest accuracy of 98.47% on the DHCD dataset. Dongre and Mankar, (2015) proposed an approach for the recognition of Devanagari numerals and characters by utilizing structural and geometric features. They extracted structural features and global geometric features, resulting in a total of 81 features to represent the image. For classification, a Multi-Layer Perceptron Neural Network (MLP-NN) was employed. The training and testing were conducted using 3,000 handwritten samples of Devanagari numerals and 5,375 handwritten samples of Devanagari alphabetic characters. The experimental results demonstrated a recognition accuracy of 93.17% for numerals using 40 hidden neurons and a recognition accuracy of 82.7% for characters using 60 hidden neurons.

Shelke and Apte, (2016) focused on optimizing the performance of neural networks for handwritten Devanagari character recognition. They employed optimized feature extraction techniques to extract structural features from the characters. The authors collected a database of 40,000 samples, which was divided into training (60% samples), validation (20% samples) and testing (20% samples) sets. They conducted a comparative analysis of different classifiers and achieved recognition rates of 97.20% with Feed Forward Back Propagation Network (FF-BPN), 97.46% with Cascade-Forward BPN (CF-BPN), and 98.10% with Elman BPN (E-BPN) after optimization. To address the challenge of shape similarity, Bhattacharya et al., (2018) introduced a novel approach called Sub-stroke-wise Relative Feature (SRF) for the recognition of online Devanagari cursive words. By utilizing Hidden Markov Model (HMM) classification and a dataset comprising 29,900 words, they achieved a word recognition accuracy of 88.09%. The proposed approach specifically targeted the challenges posed by the cursive nature of Devanagari handwriting, showcasing its effectiveness in accurately recognizing handwritten cursive words.

Jangid and Srivastava, (2018a) proposed a system for recognizing handwritten Devanagari characters based on Deep Convolutional Neural Networks (DCNN) and adaptive gradient methods. The maximum recognition accuracies obtained were 96.02% and 97.30% on the ISIDCHAR database, 96.45% and 97.65% on the V2DMDCHAR database, and 96.53% and 98.00% on the combined databases (ISIDCHAR+V2DMDCHAR) using DCNN and Layer-wise DCNN, respectively, with NA-6 and RMSProp optimizer. Jangid and Srivastava, (2018b) aimed to minimize classification errors in the recognition of handwritten Devanagari characters by identifying the critical regions and generating additional features. They employed the Fisher linear discriminant model to detect these critical regions and extract the corresponding additional features. The proposed method utilized gradient-based features and an SVM classifier for character recognition. Experimental results on the ISIDCHAR dataset, which consists of 36,172 handwritten Devanagari characters, demonstrated a recognition accuracy of 96.58%.

Gupta and Bag, (2019) proposed a novel approach for script-independent character segmentation of handwritten text. Their method utilized the structural properties of languages and employed polygonal approximation to obtain Digitally Straight line Segments (DSS) of the word. The approach was tested on four popular Indian languages, achieving an average success rate of 90.07% for character segmentation. Authors also achieved character recognition accuracies of 95.10%, 95.57%, 96.09%, and 94.71% for the Hindi language using Random Forest (RF), Support Vector Machine (SVM), Multi-Layer Perceptron (MLP) and Convolutional Neural Network

(CNN) classifiers, respectively. (Narang et al., 2019a) investigated the recognition of Devanagari ancient manuscripts. They utilized statistical features, including intersection points, open endpoints, centroid, horizontal peak extent and vertical peak extent features. Multiple classifiers, such as CNN, NN, Multilayer Perceptron, RBF-SVM and random forest techniques, were employed in their analysis. With a database of 6,152 samples, the authors achieved a recognition accuracy of 88.95% by combining various features and classifiers. In a similar vein, Narang et al., (2019b) presented a paper on recognizing Devanagari ancient manuscripts using AdaBoost and Bagging techniques. They achieved recognition accuracies of 90.70% and 91.70% using DCT zigzag features with an RBF-SVM classifier and adaptive boosting with RBF-SVM, respectively, for a database of 5,484 samples.

Narang et al., (2020) presented improved recognition results for Devanagari ancient characters by utilizing the Scale-Invariant Feature Transform (SIFT) and Gabor filter feature extraction techniques. The classification task was performed using a Support Vector Machine (SVM) classifier. The authors collected a database of 5,484 samples of Devanagari characters from various ancient manuscripts in libraries and museums. Principle Component Analysis (PCA) was employed to reduce the length of the feature vector, resulting in reduced training time and improved recognition accuracy. The proposed system achieved a recognition accuracy of 91.39% using tenfold cross-validation and a poly-SVM classifier.

Pande and Jha, (2021) presented a character recognition system for Devanagari script using a range of machine learning classifiers. They extensively investigated the performance of several classifiers, including Decision Tree (DT), Nearest Centroid (NC), K-Nearest Neighbors (KNN), Extra Trees (ET) and Random Forest (RF) classifiers. Through their analysis, the authors found that the Extra Trees and Random Forest classifiers outperformed the other classifiers in terms of Devanagari character recognition. They achieved recognition accuracies of 78% and 77% respectively, highlighting the effectiveness of these classifiers in the task at hand. Kumar et al., (2022) focused on developing a system for the recognition of hollow Hindi characters. The authors employed a range of features including zoning, horizontal projection, vertical projection, Oriented Fast and Rotated BRIEF (ORB), Scale Invariant Feature Transform (SIFT) and Speeded Up Robust Feature (SURF) to extract relevant information from the hollow Hindi character images. For the recognition task, they experimented with different classifiers such as k-Nearest Neighbor (k-NN), Support Vector Machine (SVM) and Random Forest (RF). Their dataset consisted of 3,900 distorted Hindi characters. Through their experiments, the authors achieved a maximum recognition accuracy of 91.1%, showcasing the effectiveness of their approach in accurately identifying hollow Hindi characters.

Omayio et al., (2023) presented an efficient and robust recognition system for offline handwritten Hindi characters. Their proposed approach utilized the Integral Histogram of Oriented Displacement (IHOD) for feature extraction. Authors achieved a recognition accuracy of 97.45% on a corpus consisting of 43,572 instances of words (Hindi) from 415 different writers. Their results demonstrate the effectiveness of the proposed systems for offline handwritten Hindi character recognition. Gaikwad et al., (2023) proposed a approach for the recognition of handwritten Devanagari characters. Their approach involved entropy-based skew correction for correcting skew in the characters and Mask-based algorithm for removing the header line. Authors extracted the Histograms of Oriented Gradients (HOG) features and explored AdaBoost approach for classification. They achieved a recognition accuracy of 98.43% (achieved on the V2DMDCHAR dataset) and 98.68% (achieved on the ISIDCHAR dataset) for their work.

Table 2.2 displays the recognition outcomes obtained by different researchers in the field of Devanagari script for isolated character recognition, employing distinct features and classifiers.

**Table 2.2:** Recognition results of isolated characters

| Authors | Script or Language | Dataset (Characters) | Approach | | Recognition Accuracy (%) |
|---|---|---|---|---|---|
| | | | Feature Extraction | Classification | |
| Hanmandlu et al., (2007) | Hindi | 4,750 | Box approach | Coarse | 90.65% |
| Pal et al., (2008) | Devanagari | 36,172 | Gradient and curvature | MQDF and SVM | 95.13% |
| Agrawal et al., (2009) | Hindi | 100 samples of each character | Similarity measure | Coarse | 97.25% |

| | | | | | |
|---|---|---|---|---|---|
| Kubatur et al., (2012) | Devanagari | 2,760 | DCT | NN based | 97.20% |
| Yadav et al., (2013) | Hindi | 1,000 characters and 15 paragraphs consisting of 650 words | Histogram of projection *(Mean distance, pixel value and vertical zero crossing)* | ANN | 98.50% *(characters)* 90.00% *(words)* |
| Kale et al., (2013) | Devanagari | 27,000 *(12,000 basic; 11,250 compound and 3,750 split comonent of compound characters)* | Legendre moment | ANN | 98.25% *(basic characters)* 98.36% *(compound characters)* |
| Dixit et al., (2014) | Devanagari | 2,000 | Wavelet-based | BPNN | 70.00% |
| Jangid and Srivastava, (2014) | Devanagari | 36,172 *(ISIDCHAR)* 20,305 *(V2DMDCHAR)* | GLAC | SVM | 93.21% *(ISIDCHAR)* 95.21% *(V2DMDCHAR)* |
| Acharya et al., (2015) | Devanagari | 92,000 *(DHCD)* | Deep CNN | Deep CNN | 98.47% |
| Dongre and Mankar, (2015) | Devanagari | 5,375 | Structural and Geometric | MLP-NN | 82.70% |
| Shelke and Apte, (2016) | Devanagari | 40,000 | Structural | (a) FF-BPN (b) CF-BPN (c) E-BPN | (a) 97.20%, (b) 97.46% (c) 98.10% |
| Bhattacharya et al., (2018) | Devanagari | 29,900 | SRF | HMM | 88.09% |
| Jangid and Srivastava, (2018a) | Devanagari | 56,477 *(ISIDCHAR + V2DMDCHAR)* | RMSProp adaptive gradient | DCNN | 98.00% |
| Jangid and Srivastava, (2018b) | Devanagari | 36,172 *(ISIDCHAR)* | Gradient-based | (a) KNN (b) SVM | (a) 88.90% (b) 95.37% |
| Gupta and Bag, (2019) | Hindi | 3,000 | Shadow and cumulative stretch feature | (a) RF (b)SVM (c) MLP | (a) 95.10% (b) 95.57% (c) 96.09% |
| Narang et al., (2019a) | Devanagari | 6,152 | intersection points, open endpoints, centroid, horizontal peak extent and vertical peak extent features | CNN, NN, Multilayer Perceptron, RBF-SVM and random forest techniques | 88.95% |

| | | | | | |
|---|---|---|---|---|---|
| Narang et al., (2019b) | Devanagari | 5,484 | DCT zigzag | RBF-SVM | 90.70% |
| Narang et al., (2020) | Devanagari | 5,484 | SIFT and Gabor filter | Poly-SVM | 91.39% |
| Pande and Jha, (2021) | Devanagari | 48,000 | Profiles, skeletons, contour-based | (a) DT<br>(b) NC<br>(c) KNN<br>(d) ET<br>(e) RF | (a) 63.51%<br>(b) 68.02%<br>(c) 75.14%<br>(d) 78.19%<br>(e) 76.82% |
| Kumar et al., (2022) | Hindi | 3,900 | Zoning, horizontal projection, vertical projection, ORB, SIFT and SURF | (a) KNN<br>(b) SVM<br>(c) RF | (a) 74.51%<br>(b) 87.07%<br>(c) 91.10% |
| Omayio et al., (2023) | Hindi | 43,572 | Integral Histogram of Oriented Displacement (IHOD) | MLP | 97.45% |
| Gaikwad et al., (2023) | Devanagari | V2DMDCHAR and ISIDCHAR | Histograms of Oriented Gradients (HOG) | AdaBoost | 98.43% *(V2DMDCHAR)*<br>98.68% *(ISIDCHAR)* |

Based on the information presented in the tables (refer Table 2.1 and 2.2), it can be observed that the recognition rate or accuracy of OCR systems is significantly influenced by the size of the dataset used in the experiments, as well as the choice of feature extraction methods and classification techniques employed.

### *2.2.2.4 Script Recognition*

Ghosh et al., (2010) conducted a comprehensive review of structure and visual appearance-based methods for script identification, examining the works of various researchers in detail. Pal et al., (2012) developed a lexicon-driven method for multi-lingual city name recognition written in English, Hindi and Bangla for Indian postal automation. They explored slant correction, water reservoir and Dynamic Programming (DP) concept for the recognition above mentioned languages. They achieved overall 92.25% of recognition accuracy on a corpus of 16,132 Indian city names (trilingual). Rani et al., (2014) investigated the effectiveness of Gabor filter banks combined with K-Nearest Neighbors (KNN), Support Vector Machine (SVM) and Probabilistic Neural Network (PNN) classifiers for script identification at the line level in trilingual documents. The authors achieved a high recognition rate of 99.85% for trilingual documents (Gurmukhi, Hindi and English) using Gabor features with an SVM classifier. Singh et al., (2015) discussed the advancements reported in the literature over the past few decades regarding Offline Script Identification (OSI) of Indic scripts, focusing on various feature extraction and classification techniques. Shi et al., (2016) introduced Discriminative Convolutional Neural Network (DiscCNN), a novel deep learning-based method for script identification in natural images. The method utilized deep features and discriminative mid-level representations and included the creation of a large-scale dataset called SIW-13, comprising 16,291 wild text images in 13 scripts for evaluation purposes.

Obaidullah et al., (2017) developed a word-level document image dataset consisting of 39,000 words from 13 different Indic languages, each language containing 3,000 words from 11 official scripts. The authors employed five different classifiers (namely MLP, FURIA (Fuzzy Unordered Rule Induction Algorithm), SL (Simple Logistic), LibLINEAR (Library for LINEAR classifier) and BayesNet (Bayesian Network)) and three features (namely Spatial Energy, Wavelet Energy and Radon Transform), including their combinations, for baseline results on script identification. MLP provided the best results when all features were used, achieving a bi-script accuracy of 99.24% (keeping Roman common), 98.38% (keeping Devanagari common), and a tri-script accuracy of 98.19% (keeping both Devanagari and Roman common). Furthermore, Singh et al., (2017) presented a comprehensive script recognition system for handwritten mixed-script documents. The system involved segmenting the document pages into text-lines and words, followed by word-level script recognition using texture-based features. The technique was applied to 100 mixed-script document pages containing Bangla or Devanagari text mixed with English words, resulting in encouraging outcomes.

Gupta and Bag, (2019) developed a novel approach for script independent character segmentation of handwritten text. Their method utilized basic structural properties of the languages to effectively segment characters. Experiments are carried out on the corpus of comprising four Indian languages namely Hindi, Marathi, Punjabi and Bangla. They achieved the average success rate of 90.07% for character segmentation across the four languages. Authors explored shadow and cumulative stretch based features in combination with RF, SVM, MLP and CNN classifiers for character recognition. They concluded that their proposed approach yielded improved accuracy for character segmentation as well as for recognition. Magotra et al., (2020) investigated various text segmentation algorithms based on the structural characteristics of the Takri Script for identification and classification purposes. They examined segmentation techniques for Gurmukhi characters, including touching and connected component segmentation approaches, to assess their suitability for the Takri script. Using a Naive-Bayesian classification technique, the authors achieved 80% accuracy in classifying and identifying Takri script texts. Dey et al., (2021) suggested a hybrid feature representation method for recognizing handwritten characters in Devanagari and Bangla scripts. Authors, initially extracted shape-based features of the characters, specifically focusing on angular motion, center to the thin text and center to edge text. These extracted features were then used as inputs for numerous machine learning algorithms, including two modified NN models. They concluded that their modified NN models exhibited efficient execution times when applied to the character datasets.

Malakar et al., (2022) developed a model that integrated shape transformation characteristics with a majority voting mechanism for recognizing handwritten words in Arabic and Roman scripts. They also considered inter-segment similarity to improve recognition results, outperforming cutting-edge holistic word recognition techniques. Jindal and Ghosh, (2023) proposed a novel method for segmenting textlines of ancient handwritten Devanagari and Maithili documents into words and characters. The method utilizes a three-zone division approach, including upper, middle, and lower zones. The middle zone is extracted using a linear regression curve that passes through the middle region of the textline. The proposed method is evaluated on self-generated datasets containing ancient handwritten textline images in Devanagari and Maithili scripts. The results demonstrate high accuracies of 96.31% (word segmentation) and 98.35% (character segmentation) for ancient Devanagari documents. Similarly, for ancient handwritten documents in the Maithili script, the proposed method achieves accuracies of 97.39% (word segmentation) and 98.65% (character segmentation). Moudgil et al., (2023) proposed a method for the ancient manuscript recognition written in Devanagari script using CapsNet (Capsule Neural Network). Authors divided the whole dataset into 399 classes so as to recognize basic, modifiers and conjunct characters. Their experiment, results 94.6% of recognition accuracy using CapsNet for recognizing ancient manuscript. Sharma et al., (2023) conducted a comparative analysis of two models, namely CNN based EfficientNet B3 and YOLO v4, for text recognition in the Gujarati script. Through their experimental work, they concluded that EfficientNet B3 model outperformed the YOLO v4 model in terms of both accuracy and efficiency on the collected images of Gujarati newspaper articles.

## 2.3 COMPARATIVE STUDY

In Table 2.3, a comparative study of recognition results for Devanagari handwritten character recognition in terms of accuracy (%) have been presented using the same features extraction methods with dataset and classification methods considered.

**Table 2.3:** Feature wise brief summary of handwritten character recognition

| # | Authors | Classification Techniques | Dataset | Recognition Accuracy (%) |
|---|---|---|---|---|
| **Structural Features** | Arora et al., (2007) | Feedforward Neural Network (FFNN) | 50, 000 | 89.12% |
| | Ghosh and Roy, (2015b) | Support Vector Machine (SVM) | 10,000 | 90.63% |
| | Shelke and Apte, (2015) | Multistage *(Fuzzy Inference System and Structural Parameters)* | 40,000 | 96.95% |
| | Shelke and Apte, (2016) | (a) FF-BPN<br>(b) CF-BPN<br>(c) E-BPN | 40,000 | (a) 97.20%,<br>(b) 97.46%<br>(c) 98.10% |
| | Pande and Jha, (2021) | (a) DT<br>(b) NC<br>(c) KNN<br>(d) ET<br>(e) RF | 48,000 | (a) 63.51%<br>(b) 68.02%<br>(c) 75.14%<br>(d) 78.19%<br>(e) 76.82% |
| **Statistical Features** | Sharma et al., (2006) | Quadratic | 11,270 | 80.36% |
| | Hanmandlu and Murthy, (2007) | Fuzzy | 4,750 | 90.65% |
| | Deshpande et al., (2008) | Regular Expressions (RE) & MED | 5, 000 | 82.00% |
| | Mane and Ragha, (2009) | Elastic Matching | 3,600 | 94.91% |
| | Kale et al., (2013) | Feedforward Neural Network (FFNN) | 27,000 | 98.36% |
| | Narang et al., (2019a) | CNN, NN, MLP, RBF-SVM and RF | 6,152 | 88.95% |

| | | | | |
|---|---|---|---|---|
| Gradient Features | Pal et al., (2007b) | Quadratic | 36, 172 | 94.24% |
| | Pal et al., (2008) | SVM and Modified Quadratic Discriminant Function (MQDF) | 36, 172 | 95.13% |
| | Pal et al., (2009) | Mirror Image Learning (MIL) | 36, 172 | 95.19% |
| | Kumar, (2009) | (a) SVM<br>(b) MLP | 25,000 | (a) 94.10%<br>(b) 91.90% |
| | Jangid and Srivastava, (2014) | Support Vector Machine (SVM) | 36,172 *(ISIDCHAR)*<br>20,305 *(V2DMDCHAR)* | 93.21% *(ISIDCHAR)*<br>95.21% *(V2DMDCHAR)* |
| | Jangid and Srivastava, (2016) | Support Vector Machine (SVM) | 36,172 *(ISIDCHAR)* | 96.58% |
| | Jangid and Srivastava, (2018a) | DCNN | (a) 36,172<br>(b) 20,305<br>(c) 56,477 | (a) 97.30%<br>(b) 97.65<br>(c) 98.00% |
| | Jangid and Srivastava, (2018b) | (a) KNN<br>(b) SVM | 36,172 *(ISIDCHAR)* | (a) 88.90%<br>(b) 95.37% |
| | Omayio et al., (2023) | Multi-Layer Perceptron (MLP) | 43,572 | 97.45% |
| | Gaikwad et al., (2023) | AdaBoost | V2DMDCHAR and ISIDCHAR | 98.43% *(V2DMDCHAR)*<br>98.68% *(ISIDCHAR)* |
| Global Transform | Kubatur et al., (2012) | Artificial Neural Network (ANN) | 2,760 | 97.20% |
| | Dixit et al., (2014) | Back Propagation Neural Network (BPNN) | 2,000 | 70.00% |
| | Narang et al., (2019b) | RBF-SVM | 5,484 | 90.70% |
| | Narang et al., (2020) | Poly-SVM | 5,484 | 91.39% |
| Structural and Statistical Features | Arora et al., (2009) | Multi-Layer Perceptrons (MLP) | 1, 500 | 89.58% |
| | Arora et al., (2010) | MLP and Combinational | 4,900 | 90.74% |
| | Pant et al., (2012) | Radial Basis Function (RBF) | 7,380 | 80.25% |
| | Yadav et al., (2013) | ANN | 1,000 | 98.50% |
| | Dongre and Mankar, (2015) | MLP-NN | 4,300 | 82.70% |
| | Ghosh and Roy, (2015a) | Support Vector Machine (SVM) | 5,000 | 84.10% |
| | Ghosh and Roy, (2015b) | Support Vector Machine (SVM) | 10,000 | 85.10% |

| | | | | |
|---|---|---|---|---|
| **Multi-Features** | Kumar et al., (2018) | Raw, Convex, Curvature and Writing Direction | 3,750 | 79.46% *(Lexicon based)* 71.86% *(ROVER combination)* |
| | Gupta and Bag, (2019) | (a) RF (b) SVM (c) MLP | 3,000 | (a) 95.10% (b) 95.57% (c) 96.09% |
| | Kumar et al., (2022) | (a) KNN (b) SVM (c) RF | 3,900 | (a) 74.51% (b) 87.07% (c) 91.10% |
| | Sharma et al., (2023) | EfficientNet B3 and YOLO v4 | 81,304 | 98.92% |
| **Deep Features** | Shelke and Apte, (2010) | Multistage *(Neural Network and Template Matching)* | 16,000 | 95.40% |
| | Acharya et al., (2015) | Deep CNN | 92,000 *(DHCD)* | 98.47% |
| | Deore and Pravin, (2020) | A two-stage VGG16 deep learning model | 5,800 | 96.55% |
| | Narang et al., (2021) | Convolutional Neural Network (CNN) | 5,484 | 93.73% |
| | Mishra et al., (2021) | ResNet | 92,000 *(DHCD)* | 99.72% |
| | Prashanth et al., (2022) | (a) CNN (b) MLCNN (c) ACNN | 38,750 | (a) 96.00% (b) 99.00% (c) 99.00% |
| | Pande et al., (2022) | Convolutional Neural Network (CNN) | 46-classes | 99.13% |
| | Deore, (2022) | ResNet | In-house dataset | 86% |
| | Ansari et al., (2022) | Support Vector Machine (SVM) | Real-word dataset | 99.41% |

#*Feature Extraction*

The present study does not undertake an evaluation of the effectiveness of different methods due to the absence of experiments conducted on a standardized dataset or benchmark. However, the findings of this study highlight the limited extent of research on handwritten character recognition systems with high accuracy rates in Devanagari scripts, thus indicating a potential avenue for future investigation.

### 2.3.1 Research Gaps

The field of optical Devanagari character or word recognition presents several research gaps that offer potential directions for future investigation. These include:

- Recognition of handwritten mathematical expressions remains a challenging area within character recognition, requiring further research.
- Character segmentation poses additional challenges due to issues such as overlapping, touching, and broken characters.
- Identifying optimal segmentation points for lines, words, and isolated characters is desirable yet challenging, as incorrect segmentation can lead to inaccurate recognition. The diversity of writing styles further complicates this task.
- Non-uniform backgrounds can adversely impact recognition results, posing a significant challenge to overcome.
- Recognition of historical documents presents difficulties due to factors such as poor document quality, the presence of non-standard alphabets, and unknown fonts.
- Similarity in shape between various characters in the Devanagari script, such as क-फ, ख-स, घ-ध, थ-य, ब-व, भ-म, and प-ष, contributes to misclassification. Researchers face the challenge of identifying critical regions that distinguish these similar-shaped characters.
- Artistic text, characterized by nonlinear shapes such as circles, triangles, curves, and arcs, poses a challenge for existing character recognition systems. Developing conversion models to transform artistic text into simpler linear text would enable successful character recognition in such scenarios.
- Designing classifiers and obtaining sufficient training samples become increasingly challenging as the size of the class space expands.
- No universal approach exists that is suitable for all types of documents, including degraded or historical documents in various environments.

These research gaps highlight areas where further investigation is needed to address the challenges and advance the field of optical Devanagari character or word recognition.

### 2.3.2 Deep Learning based Approaches and Research Challenges

Deep learning-based methodologies have shown promise in various areas of pattern recognition, including character recognition Deore and Pravin, (2020). These approaches offer significant potential for addressing complex tasks such as feature extraction and classification, owing to their ability to adjust the structure and parameters of deep learning models. Despite their potential to surpass traditional methods, deep learning-based approaches present ongoing research challenges Weng and Xia, (2020):

- Determining the optimal number of network layers and neurons in deep learning models is a challenging task.
- The accuracy of deep learning models heavily relies on the availability of a large dataset, necessitating the acquisition of substantial training samples.
- Selecting the optimal parameters for deep learning network architectures is another research challenge, given the various parameters involved.
- Developing efficient deep learning models that minimize memory space, computational calculations, and bandwidth requirements is a demanding endeavor.

Nowadays, developing a character recognition framework using a deep learning approach is still worth exploring.

## 2.4 DISCUSSION

This chapter provides an overview of feature extraction and classification methods employed in online and offline Handwritten Character Recognition (HCR) or Handwritten Word Recognition (HWR) systems for various scripts including Devanagari script. The evaluation of recognition accuracy results is challenging due to varying constraints, dataset sizes and sample spaces. Additionally, there is a lack of assessment tools to evaluate the performance of individual stages and overall system performance. Trade-offs between data acquisition quality and method complexity impact recognition accuracy. While significant progress has been made in HCR/HWR for Devanagari, machines still struggle to match human fluency in recognizing handwriting. Existing methods often fail to capture the nuances of handwriting generation and the perceptual process of reading. Furthermore, a standardized database for Indic scripts, including Devanagari, is lacking. Challenges are identified, suggesting future research directions, including the exploration of multi-feature combinations and the adoption of deep learning approaches. Efficient feature extraction methods focusing on shape characterization rather than color, texture or edge information are needed. Recognition of handwritten compound characters and higher-level units such as words and phrases remains a nascent area.

The presence of diverse writing styles, variations in paper quality, and the occurrence of unusual ligatures in adjacent characters pose challenges for accurate recognition of words in the Devanagari script, such as “झ”, “ड” and “इ”. There is a pressing need for further research in the area of word, sentence, and document recognition in Devanagari, including the incorporation of semantic and lexical analysis. To enhance recognition accuracy, future work should focus on developing more effective segmentation techniques capable of handling overlapping and faint characters. Additionally, the proposal and development of novel feature extraction and classification techniques are required to distinguish between highly similar and confusing characters, thereby improving the recognition rate.