"Science is a way of thinking much more than it is a body of knowledge"
- Carl Sagan

Chapter 1

INTRODUCTION

1.1 OVERVIEW

Technology has made machine-human interface extremely important in the modern day. This interface results in pattern recognition, artificial intelligence, cognitive computing, search engine and social media more user-friendly and interactive. Optical Character Recognition (OCR) is the part of pattern recognition that attempts to build and design a computer system that can convert scanned images of machine printed or handwritten text (numerals, letters and symbols), into corresponding ASCII (American Standard Code for Information Interchange) characters or machine-readable digital form, automatically (Kaur and Kumar, 2018) . It is generally used to improve the speed of operation, reduce errors or noise in the documents and decrease storage space needed for paper documents. Using digitization, various types of documents such as historical papers and books can be stored in their original forms for future generations. Thereafter, these documents can be retrieved by utilizing various pattern recognition algorithms. Therefore, OCR plays crucial role in the translating the paper-based documents into their electronic (digital) counterparts.

OCR systems viz. Handwritten Character Recognition (HCR) and Handwritten Word Recognition (HWR) systems save human energy and cost by automatically translating document images into equivalent codes (Singh et al., 2022a). Most OCR systems are script specific in the sense that they can read characters written in one particular script only (Ghosh et al., 2010; Singh and Garg, 2021). In the Indian subcontinent, a number of languages are in use and an automatic recognition of printed and handwritten scripts facilitates number of applications such as airline ticket readers, automatic license plate recognition, bill processing systems, cheque reading, data classification through learning process, editing old documents, employee code reading/verification, forensic document analysis, form processing, handwritten notes reading, image document sorting, postal automation and penetrating online libraries of image documents (Chaudhuri et al., 2017; Sharma and Dhaka, 2016; Singh and Garg, 2019; Singh et al., 2022a). Researchers had done a lot of work for the recognition of numerals, characters and word of various Indic and Non-Indic scripts. In India, 23 different languages (including English) and 13 different scripts (including Roman) exist (Jayadevan et al., 2011; Obaidullah et al., 2016). In general, OCRs are script specific, and processing documents having more than one script is not easy. But till now a very few work is available in literature for the recognition of offline handwritten Devanagari words. Devanagari script is used to write various languages including Hindi which is the widely used language of India (Singh et al., 2022a). The objective of this work is to present a framework for offline handwritten Devanagari word recognition so as to interpret the handwritten city names written on the postal envelopes.

1.2 HANDWRITTEN WORD RECOGNITION

Handwriting has been used for communication and record keeping in both modern and ancient world (Narang et al., 2021). Nowadays, one interesting, complex and challenging field in OCR is Handwritten Word Recognition (HWR) due to various writing styles of different individuals. As the handwritten word is mixture of cursive and non-cursive parts so word recognition is significantly difficult. Handwriting recognition has been the topic of research for many decades (Liwicki et al., 2014). It is a process in which hand-printed (graphical) expressions of a language are transformed into their symbolic representation (Kumar, 2009). The accuracy of HWR system highly depends upon the feature extraction and classification methods considered. Many researchers had proposed variety of feature extraction and classification methods for various scripts including Devanagari. In literature, handwritten word recognition has two streams (Dasgupta et al., 2016; Madhvanath and Govindaraju, 2001) namely online handwritten word recognition and offline handwritten word recognition as presented in Fig. 1.1. Offline handwritten word recognition approaches can further be classified into two categories namely analytic (segmentation-based) and holistic (segmentation-free) approaches.

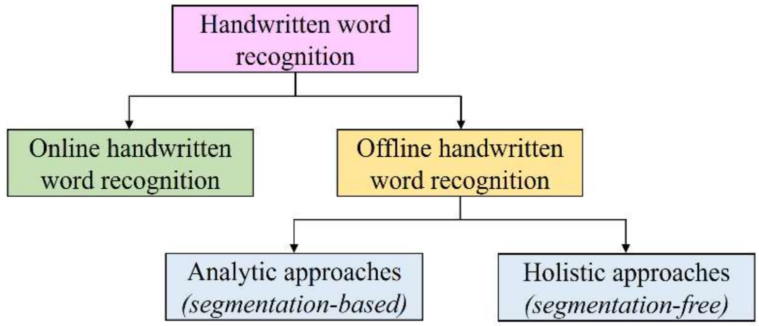

Figure 1.1: Streams of offline handwritten word recognition

1.2.1 Online Handwritten Word Recognition

In online HWR, one writes on an electronic surface such as an electronic tablet with a special pen and words or characters are recognized at real time as soon as it is written. Words or characters are captured as a sequence of strokes in these systems (Oval and Shirawale, 2015; Sen et al., 2020). Online systems obtain the position of the pen as a function of time in the form of (x, y) coordinates, directly from the interface. The recognition algorithms may therefore access the temporal information, such as the position and velocity of the pen together with its track. It is also known as ‘real time’ handwritten word recognition because the majority of algorithms try to read the text as it is being written.

1.2.2 Offline Handwritten Word Recognition

It is the process of translating offline handwritten word into a format that is understood by the machine. In offline HWR, the handwritten word is typically scanned from a paper document and made available in the form of a binary or gray scale image to the recognition algorithm. Offline HWR become more challenging due to shape of characters, great variation of character symbols and document quality (Su et al., 2015). Therefore, offline word or character recognition is considered as a more challenging task than its online counterpart. Offline HWR is significantly different from online HWR as depicted in Table 1.1. It demonstrates the recognition accuracy and speed of recognition are the main advantages of online handwritten word recognition over offline handwritten word recognition. Therefore, it is obviously that offline handwritten word recognition becomes a challenging task.

Table 1.1: Offline versus online handwritten word recognition

Parameters	Offline HWR	Online HWR
Availability of strokes	No, there is non-availability of the stroke's information.	Yes, there is availability of the stroke's information.
Process	It is an offline process that captures the static information of handwritten words after it is written on the paper.	It is an online process that captures the dynamic information of handwritten words during the time when it is actually written (real-time) on electronic surface.
Raw data requirement	Dots per inch.	Samples per second.
Writing media	Paper document.	Digital pen and an electronic surface.
Pre-processing	It requires more preprocessing operations.	It requires less preprocessing operations.
Recognition speed	Recognition speed is lower.	Recognition speed is sufficiently higher.
Recognition accuracy	It supports relatively less accurate recognition.	It supports sufficiently more accurate recognition.
Suitability	It can be suited for already handwritten papers.	It cannot be used for already handwritten papers.

Researchers used several approaches/methodologies for the recognition of offline handwritten Devanagari words. Generally, these approaches/methodologies can be categorized into three categories as briefly mentioned in the following subsections (Kumar, 2016).

1.2.2.1 Segmentation Based or Analytical Approach

In this approach, the given word is segmented into individual components (or character parts or even character subparts) and each component (or character) is recognized and assigned a symbol and the resulting symbols are reconstructed (or reassembled) to know the identity of a word. Recognition is considered as a matching process which is usually done with the help of dictionary/library words (Garg et al., 2015a; Shearme and Leach, 1968). The segmentation for obtaining each individual character is usually based on heuristics and done before the recognition of each character.

1.2.2.2 Segmentation Free or Holistic Approach

It is a word based approach, where a word is not broken down into individual components (or character parts). The word is treated as a single unit (or a whole) and there is no attempt to identify characters individually (Dasgupta et al., 2016). Classifier is trained with a complete set of words by extracting their properties that helps to recognize the word as a whole. Challenges to this approach include complexity as the whole word is treated as a single indivisible entity and lower discrimination capabilities. The use of holistic approaches is usually restricted to areas where few word groups are used (small size vocabulary) and constrained to a fixed lexicon such as numbered words on cheques. The characters in a word may touch or overlap with each other as and when these are writing on paper, which causes problems with word segmentation. Using above mentioned approach of word recognition, the problem of segmentation can be avoided. These approaches have been less studied than part-based ones and recently have gained considerable interest.

For the past few years, researchers have been working really hard in this sector (Adak et al., 2016; Bhowmik et al., 2019; Bhowmik et al., 2014; Bhunia et al., 2018; Dasgupta et al., 2016; Garg et al., 2015b; Jayadevan et al., 2011; Kaur and Kumar, 2018, 2021a; Kumar, 2016; Malakar et al., 2017; Pal et al., 2009b; Parui and Shaw, 2007; Patil and Ansari, 2014; Ramachandrula et al., 2012; Roy et al., 2016; Shaw et al., 2014; Shaw et al., 2015; Shaw and Parui, 2010; Shaw et al., 2008a). In India, less than 10% of newspapers are read in English, 33% of newspapers are read in Hindi, and the remaining newspapers are read in regional languages, according to the 2012 national readership survey. There are 13 different scripts and 22 regional languages that are used in India (Obaidullah et al., 2017). Therefore, to deliver services to the users of these regional languages, there is the need to develop the offline handwritten recognition systems for regional characters/words such as Devanagari, Gurumukhi, Bangla etc. The work described in the thesis is an effort in this direction so as to develop a recognition system that can recognize offline handwritten words written in Devanagari script in holistic manner.

Moreover, handwritten documents can be found at a variety of places including schools/colleges/universities, banks, post-offices, etc. These places typically contain a significant amount of handwritten data, such as signatures, fax numbers, postal addresses, etc. So, the research work carried out in this direction shall be significant for above mentioned cases/scenarios.

1.2.2.3 Hybrid Approach

In hybrid, the combination of segmentation based and holistic approaches is used (Singh and Garg, 2021).

1.3 APPLICATIONS AND CHALLENGES

1.3.1 Applications of Handwritten Word Recognition

Handwritten word recognition is a crucial component of optical character recognition (OCR) systems that plays a vital role for converting printed or handwritten text into a machine-readable format. It facilitates automated processing and analysis of textual data. Some noteworthy applications of handwritten word recognition in the OCR field are listed below (Jayadevan et al., 2011; Kaur and Kumar, 2018; Singh and Garg, 2019, 2021; Singh et al., 2022a; Singh et al., 2022b; Ye and Doermann, 2014):

Document digitization: HWR plays a significant role in digitizing documents containing handwritten text. It facilitates the conversion of handwritten documents (such as historical manuscripts and personal notes) into digital format. Thus, it contributes for enhancing accessibility and streamlining the handling of valuable documents by empowering efficient storage, retrieval and analysis of handwritten texts.

Form processing: Numerous organizations deal with a lot of paper-based forms such as surveys, applications and questionnaires. HWR is used to automatically extract relevant information from these forms, significantly reducing manual data entry efforts and improving data accuracy.

Postal services: OCR systems are extensively used in postal services for sorting and routing mail. HWR systems can be employed to automatically interpret handwritten addresses and/or pin codes on envelopes, ensuring efficient delivery and thereby reducing errors.

Bank cheque reading: HWR can be utilized in OCR based systems to process and verify cheque by banks. It makes possible to extract the key information from the handwritten sections/portions of the cheque, such as the payer's name, payment amount and account number.

Handwritten text analysis: It allows researchers extract statistical data and/or analyze handwriting styles from handwritten material for various purposes such as historical research, forensic analysis and linguistic studies.

Accessibility for blind and visually impaired individuals: HWR systems can be employed in OCR systems to assist blind and visually impaired individuals. By converting handwritten texts into digital format, it becomes possible to use text-to-speech technologies or braille displays to make the content accessible to blind and visually impaired individuals.

Other applications of HWR/OCR systems include Airline ticket readers, Automatic license plate recognition, Bill processing systems, Data classification through learning process, Editing old documents, Employee code reading/verification, Forensic document analysis, Form processing, Human-robot interaction, Library archival, Meaning translation, Passport number reading/verification, Postcode/pin-code recognition for postal automation, Recognition of ancient documents, Sign board translation, Signature verification, Writer verification and Handwritten notes reading. The technology is being continuously advancing, enabling new applications and improving the accuracy and efficiency of automated text recognition.

1.3.2 Challenges of Handwritten Word Recognition

To achieve greater accuracy in the HWR system, it is crucial to have high-quality or high-resolution images that exhibit key structural properties, such as clear differentiation between text and background. The quality of the input image directly impacts the performance of the HWR system. However, achieving successful automation in HWR techniques requires overcoming various errors that can significantly degrade image quality. These errors (Arora et al., 2010b; Kaur and Kumar, 2018; Kompalli et al., 2005; Pagare and Verma, 2015; Singh et al., 2022a; Ye and Doermann, 2014) are categorized as follows:

- **Aspect ratio:** Text elements in images can vary significantly in length, with some being short while others may be much longer such as traffic signs and video captions. Detecting and processing text requires a search procedure that considers the location, scale, and length of the text. This introduces a high level of computational complexity to accurately identify and extract text from images.
- **Blurring and degradation:** Sharpness of characters in the input images are essential for improving the accuracy of character recognition and segmentation. There may be uneven focus due to small change in view or trying to capture a moving object. The resultant images become blurry and degraded as a result of this uneven focus, thus decreasing the OCR system's accuracy. To achieve best OCR performance, character representations must be kept constant and precise (Meshesha and Jawahar, 2008).
- **Character complexity:** Moreover, handwritten Devanagari characters are more complex due to their structure and shape. They include a large character set with more curves, loops, and other details in the characters.
- **Complex background:** Working over a complex background may also be a much greater challenge for the OCR system than working with normal backgrounds.
- **Different shapes and size of characters:** Segmentation and classification become a challenging task for handwritten character recognition due to the different shapes and size of handwritten characters.
- **Existence of uneven illumination:** Capturing images in natural environments may result in uneven lighting and shadows. It may further cause less accurate detection, segmentation, and recognition due to degradation of the desired characteristics of the image that introduces a challenge.
- **Lack of standard test database:** Unfortunately, little standard handwritten character database of Devanagari script is available publicly as a benchmark for

experimentation so that the effectiveness of recognition accuracies of various techniques can be compared on a common platform.

- **Larger character set is due to modifiers:** In the Devanagari script, there are upper and lower modifiers due to which two successive lines may overlap with each other. It may result in poor segmentation and hence lower recognition rate.
- **Low resolution:** Recognizing text captured in a photograph or scene text with low resolution remains an unsolved problem in OCR systems until the captured image is preprocessed with suitable preprocessing methods.
- **Noisy background:** Generally, it can be seen that noise gets added to the document/image during the scanning phase. Later, it becomes challenging to remove such background noise while performing digitization or binarization.
- **Physical and mental state of the writer:** Developing a framework for character recognition also poses a challenge to researchers due to the physical and mental state of the writer, writing instrument, pen width, ink color, and many other such factors.
- **Poor quality of documents:** These types of documents usually consist of holes, spots, noise, broken strokes, etc. which may result in the process of line segmentation very challenging.
- **Scene complexity:** Numerous man-made objects such as buildings, painting; appears in a natural environment having similar structural properties and appear as text. It imposes challenges in text recognition in the processed image making it difficult for OCR systems to distinguish text from non-text.
- **Similar-shaped characters:** Another challenge for character recognition is to recognize similar shaped characters or symbols. In Devanagari script, there exists many character pairs such as क-फ and घ-ध that are quite similar in shape.
- **Skewness:** For optical character recognition systems, the skew correction has remained a challenge (Bathla et al., 2019; Liang et al., 2008) and various researchers have proposed easier and effective processes to correct the skewness of images such as an OJ method (Obaida et al., 2011) that is suitable for any degree of rotation. Poor results may be observed if a skewed image is inputted directly into the OCR system without applying any suitable preprocessing method.

- **Speed of writing:** Characters can be represented as the trajectory drawn by the pen (up/down) on a writing medium. The nature of characters such as overlapping and touching also depends on the speed of writing which sometimes becomes a challenging problem during character recognition.
- **Variations of text layout or fonts:** Characters in cursive or italic style and script fonts of characters may cause difficulty in segmentation due to their overlapping with each other (Thakral and Kumar, 2014). It will be difficult to recognize the characters when the class number is large i.e., has large within-class variations and from many pattern sub-spaces.
- **Various styles of human writing:** Every human being has his/her own and different style of writing which may cause difficulty to recognize the characters. Character size, shape, orientation, etc. varies from person to person.
- **Warping:** For OCR systems, warping or elastic deformation of the images could be another challenge where content or characters with varying geometry have to be recognized. Such a situation may arise when an image is captured using handheld cameras. Ulges et al., (2005) and Meshesha and Jawahar, (2008) have found potential for rectification of warped document images called de-warping.
- **Limited training data:** Developing robust handwritten word recognition models requires large amounts of high-quality training data. However, obtaining a diverse and comprehensive dataset that covers various handwriting styles and variations can be challenging.
- **Multi-lingual and multi-script recognition:** Handwritten word recognition becomes more challenging when dealing with multiple languages and scripts. Different scripts have unique characteristics, and recognizing handwritten text across different languages adds another layer of complexity to OCR systems.

These challenges much be considered during the designing and implementation of the character/word recognition system to make it more effective.

1.4 PROS AND CONS

Pros and cons of handwritten word recognition have been presented in the following sub-sections.

1.4.1 Pros of Handwritten Word Recognition

Handwritten word recognition offers several advantages in the field of optical character recognition (OCR) as listed below:

- **Enhanced accessibility:** It enables the conversion of handwritten text (character or word) into machine-readable form, making it accessible to individuals with visual impairments or reading difficulties. It helps them to interact with and understand the content of handwritten documents, thereby promoting inclusivity.
- **Preservation of historical and cultural heritage:** HWR plays a crucial role in preserving and digitizing historical manuscripts and other handwritten documents by converting them into digital format. Thus, valuable artifacts can be protected from deterioration and made accessible for future generations to study and further use.
- **Automation and Efficiency:** It automates the process of converting handwritten text into machine-readable form and thereby eliminating the need for manual transcription or data entry, saving significant time and effort. It also reduces the chances of human errors that may occur during manual data entry.
- **Improved Search ability and Indexing:** OCR systems with HWR capabilities enable efficient text-based searches within handwritten documents. It shall improve the search ability and retrieval of information from vast collections of handwritten documents which will make it easier to find specific words/phrase.
- **Integration with Digital Workflows:** Once converted into machine-readable text, the handwritten text can be easily processed, analyzed, shared, and incorporated into various digital applications such as databases or data analytics tools. It shall facilitate the integration of handwritten content into digital workflows.
- **Real-Time Data Capture:** HWR can be applied in real-time scenarios, such as digital pens or tablets, allowing users to write naturally and have their handwriting recognized instantly. This enables applications like digital note-taking or interactive forms, where handwritten input is seamlessly converted into digital text.
- **Multi-lingual and Multi-script Support:** HWR based systems can handle multiple languages and scripts, accommodating the diverse needs of users

worldwide. It shall support the recognition of handwriting in various languages, enabling the digitization and processing of handwritten content from different linguistic backgrounds.

1.4.2 Cons of Handwritten Word Recognition

HWR offers numerous pros, however it also faces certain cons as listed below:

- **Recognition Errors:** HWR is prone to errors, particularly when dealing with complex handwriting styles, degraded or low-quality input, or unusual variations in letter shapes. Recognition errors can lead to inaccurate conversion of handwritten text into machine-readable form, impacting the reliability and usability of the OCR system.
- **Limited Vocabulary:** HWR models often perform better on words within their training vocabulary. Recognizing out-of-vocabulary or rare words becomes challenging, as the models may struggle to generalize well to unseen words or to words that deviate significantly from the training data.
- **Need for Training and Adaptation:** HWR systems typically require training on large datasets to achieve good performance. Additionally, they may need adaptation for specific handwriting styles or individuals, which shall add complexity and time-consuming efforts for customization.
- **Dependence on Handwriting Legibility:** Handwritten word recognition heavily relies on the legibility of the handwriting. If the handwriting is illegible, heavily stylized, messy, too faint and has severe distortions, recognition accuracy may significantly decrease. Thus, it can pose significant challenges for recognition.
- **Time and Computational Costs:** Developing accurate HWR models often requires substantial computational resources and time for training, optimization, and fine-tuning.
- **Privacy and Data Security:** Handwritten word recognition may involve the processing and storage of sensitive or personal information. Ensuring privacy and data security becomes crucial, as handwritten documents may contain personal, financial, or confidential data that needs to be protected throughout the OCR pipeline.

- **Integration Complexity:** Integrating HWR into existing OCR workflows or systems can be complex, particularly if the OCR system relies on multiple recognition modules or needs to handle various types of input, such as printed text and handwriting, simultaneously.

1.5 OVERVIEW OF DEVANAGRI SCRIPT

Devanagari belongs to the Brahmic family of scripts of India, Nepal, Tibet, and the South Asian subcontinent (Acharya et al., 2015). It is adopted by more than 500 million people and is being used for writing numerous languages viz. Hindi, Sanskrit, Marathi, Nepali including similar other languages of the South Asian subcontinent (Connell et al., 2000; Kubatur et al., 2012). The Devanagari script consists of 13 vowels, 34 consonants, and 14 modifiers of vowels as depicted in Fig. 1.2. Moreover, apart from the above, it has compound or composite characters which may be formed by combining two or more basic characters. Compound characters and modifiers can be attached adjacent to each other, on the top side, or the bottom side of the basic character (Bhattacharya et al., 2018). A vowel followed by a consonant may take a modified shape, depending on whether the vowel is placed to the left, right, top, or bottom of the consonant and are known as modifiers or matras. There is no idea of lower and upper characters and characters, including text and digits are written from left to right.

Devanagari script has its own specified composition rules for combining vowels, consonants, and modifiers (Arora et al., 2009; Jayadevan et al., 2011). An additional feature of Devanagari is the existence of a horizontal line on top of characters called a header line or shirorekha (Dixit et al., 2014; Verma and Tiwari, 2015). Two or more characters are joined to form a word by joining header lines of individual characters. A word written in Devanagari script may be divided into strips viz. top, core, and bottom. This header line divides the top and core strips whereas the virtual baseline divides the core and bottom strips. Knowledge of scripting is important in a sense, if a person knows the script of a language, then he/she can easily read the words pertaining to that script on the basis of his/her mental dictionary. Fig. 1.2 represents consonants and their corresponding half forms, vowels, and modifiers of the Devanagari script. Three strips of a word in the Devanagari script are depicted in the following Fig. 1.3.

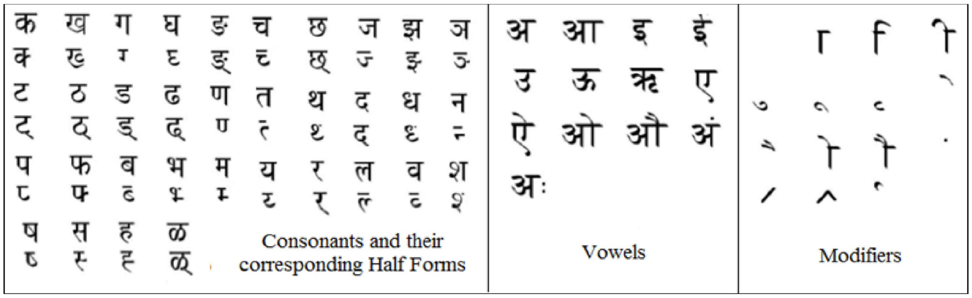

Figure 1.2: Devanagari script **(a)** Consonants and their corresponding half forms **(b)** Vowels and **(c)** Modifiers

Figure 1.3: Three strips of a word in Devanagari script

1.6 PHASES OF AN OFFLINE HANDWRITTEN WORD RECOGNITION SYSTEM

In this section, important phases (also stages or steps) of HWR system have been discussed. In general, these phases include data collection, image acquisition and digitization, pre-processing, segmentation, feature extraction and classification as depicted in Fig. 1.4.

1.6.1 Data Collection

Data collection is a crucial step in developing handwritten Devanagari word recognition systems. It involves acquiring a diverse and representative dataset of handwritten Devanagari words to train and evaluate the recognition models. The quality and size of the dataset significantly impact the performance and generalization ability of the recognition system.

1.6.2 Image Acquisition and Digitization

Digitization involves the scanning of a handwritten paper-based document to generate a digital image or electronic form, typically in the form of a bitmap. This process converts the document into a format that can be further processed during the pre-processing phase.

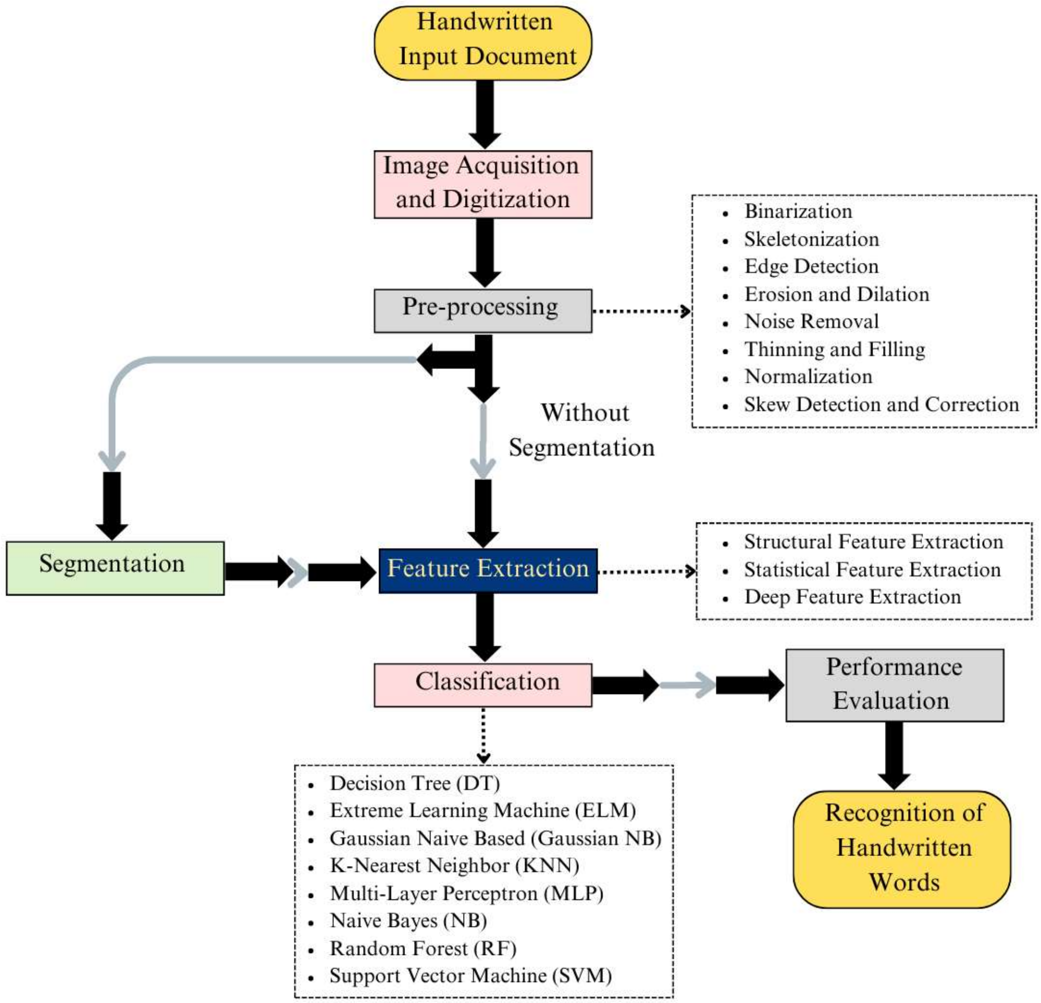

Figure 1.4: Steps for handwritten word recognition

1.6.3 Pre-processing

The pre-processing phase serves as an initial step to mitigate the degradation of the captured image and generate a normalized bitmap image. This phase encompasses various essential steps, including binarization, skeletonization, image dilation, edge detection, noise removal, contrast stretching for image enhancement, thinning and filling, normalization, as well as skew detection and correction (Alginahi, 2010; Farkya et al., 2015; Kale et al., 2013).

- **Binarization:** Binarization is the process of converting a grayscale image into a binary image with only two levels, 0 and 1. It separates foreground pixels from background pixels by applying a threshold. This technique is commonly used to distinguish objects from the background in an image.

- **Skeletonization:** Skeletonization reduces the foreground regions in a binary image to a skeletal remnant while preserving the original region's extent and connectivity. It is often employed to decrease the line width of text from multiple pixels to a single pixel.
- **Edge Detection:** Edge detection involves identifying the edges or boundaries of objects in a digitized image. Various techniques such as Sobel, Canny, and first and second derivative methods can be used to detect edges, enabling the selection of object outlines.
- **Erosion and Dilation:** After detecting the edges, erosion and dilation operations are applied to adjust the size of objects and prepare the image for segmentation. Erosion erodes away pixels on the image edges, resulting in smaller objects, while dilation adds pixels around the edges, enlarging the objects.
- **Noise Removal:** Noise removal eliminates unwanted artifacts or irrelevant bits in a document image to enhance processing. Morphological operations, filtering methods (e.g., median, Gaussian, mean, min-max, and Wiener), and noise modeling techniques can be employed to remove noise from images.
- **Thinning and Filling:** Thinning reduces the width of characters in a scanned image, improving visibility and preserving structural information. Filling, on the other hand, eliminates gaps, breaks, and holes in digitized characters. Thinning selectively removes foreground pixels related to shape information, while filling fills in the gaps (Hanmandlu and Murthy, 2007).
- **Normalization:** Normalization is applied to enhance the accuracy of Optical Character Recognition (OCR) systems. It ensures uniform character size, rotation, and slant by reducing shape variations in the scanned document. Normalization significantly reduces data size without altering the structural information of the image (Jangid and Srivastava, 2014).
- **Skew Detection and Correction:** Skewness refers to the tilt or misalignment of a bit-mapped image of a scanned document, which can be caused by human handwriting or scanning process errors. Skew detection and correction techniques are used to align such documents or images correctly. These techniques include

analyzing projection profiles, employing Hough transforms, clustering, connected component analysis, and correlation between lines.

1.6.4 Segmentation

Segmentation plays a crucial role in handwritten Devanagari word recognition systems by partitioning of a scanned document into paragraphs, lines, words, and individual characters. It enables the partitioning of a scanned document into paragraphs, lines, words, and individual characters. This step is essential for accurately recognizing and understanding handwritten Devanagari script. Segmentation techniques for Devanagari script face challenges due to complex character structures and varying writing styles. Various approaches have been proposed for segmentation in Devanagari word recognition systems. These methods often utilize features such as curvature, concavity, and connectivity to detect boundaries between characters (Kohli and Kumar, 2021; Koshti and Govilkar, 2012).

1.6.5 Feature Extraction

Feature extraction techniques plays a major role and important phase in pattern recognition (Cilia et al., 2019), that aim to extract the relevant and discriminative information from the input images of handwritten words. Features represent precise information extracted from words that distinguish a particular word from other words. Recognition accuracy of a handwritten word recognition system also depends upon the selection of feature extraction technique. Feature extraction can be carried out in number of ways, however essential is to extract those features that can distinct dissimilar patterns or word classes that exist, making it challenging for a machine learning algorithm to effectively classify and recognize words. Feature extraction techniques are briefly classified into the following subsection.

1.6.5.1 Statistical Feature Extraction

It refers to the process of extracting relevant statistical measures from the input images, to represent the characteristics of handwritten words. The statistical features serve as inputs to machine learning algorithms or other recognition models for the task of interpreting and recognizing or classifying the handwritten words. These techniques plays a crucial role by capturing meaningful information that represents the distinctive characteristics of handwritten words. Histogram of Oriented Gradients (HOG) (Bhowmik et al., 2014b; Roy et al., 2016), Zone-Based Features (Kaur and Kumar, 2021a, 2021b; Singh et al., 2022b), Statistical Moments (Kumawat et al., 2013), Local Binary Patterns (LBP) (Bahram, 2022) and Run-Length Encoding (RLE) (Bhadrannavar et al., 2020) are some examples of statistical features in handwritten word recognition.

1.6.5.2 Structural Feature Extraction

It refers to the process of extracting relevant features that capture the structural properties and spatial relationships between different components of handwritten words. These features provide information about the arrangement, connectivity, and relative positions of strokes, characters, and other components within the word image. Structural features play a crucial role in distinguishing different handwriting styles, characters, and word structures. Structural features depict a pattern in terms of its topology and geometry by giving it local and global properties. These features are mainly based on geometrical properties of a symbol or character viz. loops, directions of strokes, intersections of strokes and end points (Koshti and Govilkar, 2012; Kumar, 2016).

1.6.5.3 Deep Feature Extraction

By extracting meaningful features, the dimensionality of the input data can be reduced, and the resulting feature representation can be more informative and suitable for subsequent classification or recognition tasks.

1.6.6 Classification

Generally, after feature extraction there is another significant step called as classification. In the classification step, handwritten words are classified/recognized on the basis of the extracted features. Basically, this step identifies to which class the handwritten word belongs to. In literature, there exist a variety of classifiers to perform this task. For this work, various classifiers namely Decision Tree (DT), Extreme Learning Machine (ELM), Gaussian Naive Based (Gaussian NB), K-Nearest Neighbor

(KNN), Multi-Layer Perceptron (MLP), Naive Bayes (NB), Random Forest (RF) and Support Vector Machine (SVM) have been considered to classify handwritten Devanagari word images. The comparative analysis of proposed systems have been carried out using above mentioned classifiers to test their suitability to recognize Devanagari handwritten words. These classifiers are briefly discuss in the following subsection.

1.6.6.1 Decision Tree (DT)

The decision tree is a supervised learning based machine learning algorithm that is generally used to recognize the class of a pattern and find its application in classification problems like handwritten character/word recognition. As its name decision tree indicates, the algorithm is based on breaking down the data by asking a series of questions to make a particular decision. It has tree like graph, where a question is asked at each node. Based on the feature set received in input, the decision tree tries to match the pattern accordingly. This classifier continue its classification on a particular problem, in sub-parts until it reaches to the final conclusion. Among the n traits, one trait is chosen at root node and rest traits are set at diverse levels of tree nodes. Determination of root trait could be a tiresome assignment and it may deliver poor recognition outcomes if root node is taken as random. Refer the Fig. 1.5 as an example, where a decision tree can be used to recognize the particular class of Handwritten Devanagari Word (HDW) such as class-1, class-2 and so on.

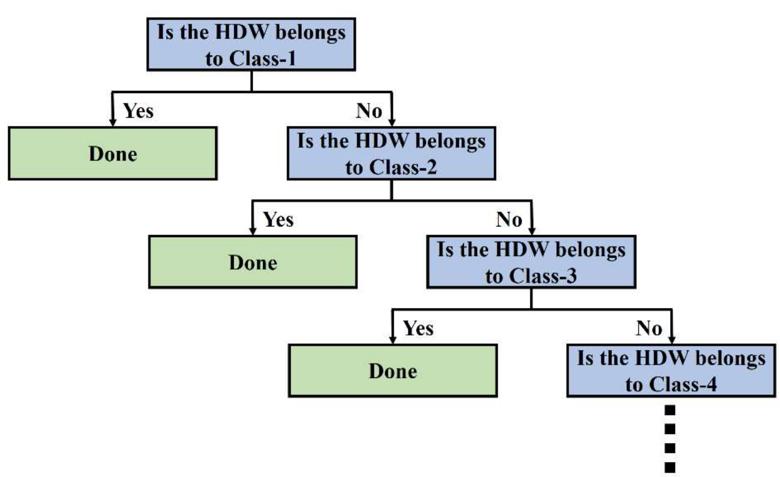

Figure 1.5: Decision Tree (DT) classification

For this work, C4.5 decision tree algorithm has been implemented developed by Quinlan as it is efficient and powerful (Budiman et al.,2017). Moreover, it has capability to deal continuous and categorical values. Mainly, decision tree C4.5 algorithm consists: choice of trait as a root, generates branch, puts dataset in branch and repeat the process until each class shall have the same value. The trait with the highest information gain value can be used as the root trait and equation for the same is as follows (Refer Eq. 1.1):

$$Information\ Gain\ (C,T) = Entropy\ (C) - \sum_{j=1}^{m} \frac{|C_j|}{|C|} \times Entropy\ (C_j) \tag{1.1}$$

Where, C represents set of case; T is a trait of the case; $|C_j|$ is the number of cases to j; $|C|$ is the number of cases in the set and $Entropy\ (C)$ can be calculated as given below (Refer Eq. 1.2):

$$Entropy\ (C) = -\sum_{j=1}^{m} p_j \times log\ p_j \tag{1.2}$$

Where, p represents the fraction of traits in a particular class. The pseudo code for decision tree C4.5 algorithm is as follows:

Input: Dataset *D* with traits *j*
Tree = { }
If *D* is "pure" or alternative stopping criteria met **then**
  stop
end if
for all traits $j \in D$ do
  Compute information-theoretic criteria if split on *j*
end for
j_{best} = Best trait according to information-theoretic criteria
T_{ree} = generate a decision node that test j_{best} in the root
D_v = simulate sub dataset from D based on j_{best}
for all D_v **do**
  T_{ree_v} = $C4.5$ (D_v)
  Append T_{ree_v} to the corresponding branch of T_{ree}
end for
return $\boldsymbol{T_{ree}}$

1.6.6.2 Extreme Learning Machine (ELM)

Extreme Learning Machine (ELM) is a training algorithm that converges far faster as compared with traditional methods and yields encouraging results (Wang et al., 2021). It learns through examples and hence can be considered as a simpler straightforward algorithm or procedure (Ghadhban et al., 2021). It does not require adjustment or updation of the parameters during the training phase as it simply updates the hidden layer nodes so as to obtain the best possible solution (Ali et al., 2020). This algorithm has the ability to prevail over the disadvantages of the backpropagation gradient method. The target output of Extreme Learning Machine (ELM) can be expressed in terms of following expression (Refer Eq. 1.3):

$$t_k^i = \sum_{j=1}^{m} \beta_{kj} g_j(\mathbb{w}, \mathbb{b}, \mathbb{x}), k = 1, 2, \ldots.., l \tag{1.3}$$

Where, $\mathbb{w}$ and β are the input and output weights respectively, $\mathbb{b}$ represents hidden layer and g_j denotes the activation function of $\mathbb{b}$ i.e. hidden layer.

Defining, H as the output matrix of the hidden layer as below (Refer Eqs. 1.4 to 1.7):

$$H(\mathbb{w}, \mathbb{b}, \mathbb{x}) = \begin{bmatrix} g(\mathbb{w}_1.\mathbb{x}_1 + \mathbb{b}_1) & \ldots & \ldots & g(\mathbb{w}_M.\mathbb{x}_1 + \mathbb{b}_M) \\ \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots \\ g(\mathbb{w}_1.\mathbb{x}_N + \mathbb{b}_1) & \ldots & \ldots & g(\mathbb{w}_M.\mathbb{x}_N + \mathbb{b}_M) \end{bmatrix}_{N \times M} \tag{1.4}$$

$$T = H\beta \tag{1.5}$$

$$\beta = [\beta_1, \beta_{2,} \ldots \ldots, \beta_M]_{1 \times M}^T \tag{1.6}$$

$$T = [t_1, t_{2,} \ldots \ldots, t_M]_{1 \times M}^T \tag{1.7}$$

β can be obtained analytically using minimum norm least square solution, as given in Eq. 1.8:

$$\tilde{\beta} = argmin_{\beta}\|H\beta\| = H^{+}T \tag{1.8}$$

Where H^{+} represents the M-P (Moore-Penrose) generalized inverse of H. In case of non-singular H, above equation can be expressed as given in Eq. 1.9:

$$\tilde{\beta} = H^{+}H^{-}H^{T}T \tag{1.9}$$

ELM tasks include recognition, prediction, representation/feature learning, clustering and surrogate modeling (Wang et al., 2021).

1.6.6.3 Gaussian Naive Based (Gaussian NB)

Gaussian Naive Bayes (Gaussian NB) is a variant of Naive Bayes which is based on Gaussian (normal) distribution. It is generally assumed that the values associated with each class are distributed in accordance with a normal or Gaussian distribution (Mansour, 2018). The likelihood of the features, can be expressed as given in the Eq. 1.10 below:

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \tag{1.10}$$

It is assumed that variance is independent of y (i.e. σ_i) or x_i (i.e. σ_k) or both (i.e. σ). This approach find its applications in creating a model that may fit by simply calculating the mean and standard deviation of the points within each class label.

1.6.6.4 K-Nearest Neighbor (KNN)

K-Nearest Neighbor (KNN) is a supervised, non-parametric, multi-functional machine learning algorithm that can be used in various application areas including handwritten character/word recognition (Abu Alfeilat et al., 2019; Tan, 2006). As its name indicates, it recognize the class of new input data point by considering 'K' nearest neighbors or feature similarity or data points. Here, 'K' represents the number of sub-feature vector that has to be stored in a feature vector as a deciding factor for recognition of a pattern/shape of unknown handwritten word. For this work, value of 'K' is taken as 1 i.e. single nearest neighbor. The algorithm computes the neighborhood or distance of an unknown input data from the target function based on the local minimum. Distance can be calculated using various distance metric such as Euclidean distance, Manhattan distance, Minkowski distance, Hamming distance. In this work, the Euclidean distance 'd' has been measured using the following Eq. 1.11:

$$d = \sqrt{\sum_{j=1}^{N} (x_j - y_j)^2} \tag{1.11}$$

Where, j represents the number of features containing in the feature set, x and y are the features stored in training data and candidate feature vector. Nearest feature vector having minimum Euclidean distance shall vote to the candidate feature for recognizing its class. It has advantage of robustness where the training data is noisy. Fig. 1.6 depicts the case of KNN classification for $K = 1$, where a red square (Class-1) is the nearest neighbor to the test sample that falls inside the dotted circle.

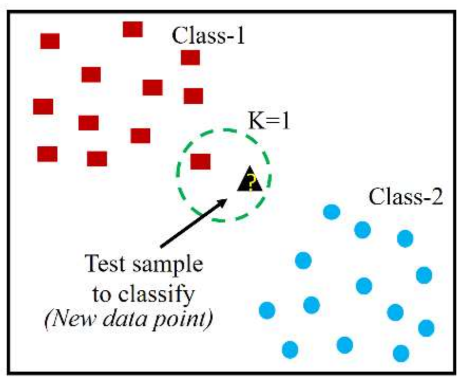

Figure 1.6: K-Nearest Neighbor (KNN) classification (K=1)

Consider, D is the dataset and an integer K represents the minimum number of near neighbors that can be taken to establish proximity. Followings are the steps to predict the output y for a new observation x:

Step-1: Load the training data.

Step-2: Organize the data by applying suitable required techniques such as scaling, missing-value treatment and dimensionality reduction.

Step-3: Find the optimal value for K. Here it is considered as $K = 1$.

Step-4: Select the K number of the neighbors.

Step-5: Predict a class value for new data:

a) Calculate the Euclidean distances between the x observable and all the data points of K number of neighbors.
b) Take the K observations that constitute the smaller distances to the observable point x.
c) Among these K neighbors, count the number of the data points in each category.
d) With the y outputs taken from the K observations, use the mode of y deductions.
e) The final prediction shall be done based on the most frequent class as calculated in above steps.

1.6.6.5 Multi-Layer Perceptron (MLP)

Multi-Layer Perceptron (MLP) is a type of feed-forward neural network that can be used for classification after suitable training (Tamen et al., 2017). It has been used in various fields of pattern recognition due to its various advantages. As there is no loop inside the network, hence information transfer takes place from input to output nodes via hidden layers. In this work, MLP is also explored for the classification of handwritten Devanagari words. To train MLP based classifier, back propagation algorithm has been employed with $\gamma = 0.3$ and $\alpha = 0.2$, similarly as given in (Kumar, et al., 2013) for handwritten Gurumukhi characters. Where, γ and α represent the learning rate and momentum, respectively.

1.6.6.6 Naive Bayes (NB)

Naive Bayes (NB) classifier is a simple and effective supervised learning algorithm that is based on Bayes hypothesis (John and Langley, 1995). It could be used to solve classification tasks including text classification (Wu et al., 2008). As prediction is carried out on the basis of the probability of a class and hence it is termed as probabilistic classifier. NB classifier is used for binary and multiclass classification problems. It assumes that all features or predictive characteristics are independent or unrelated in a given class. For recognition system, initially it builds a probability model (say class c) from extracted features of training database as per the following Eq. 1.12 (Bansal et al., 2021):

$$P(s_1, s_2, \dots \dots .., s_q | t = a) = \prod_{j=1}^{q} P(s_j | t = a) \tag{1.12}$$

Then, computation of conditional probability distribution is done for the feature vector of a class, using below equation (Refer Eq. 1.13):

$$P(t = a | s_1, s_2, \dots \dots .., s_q) = \frac{P(t = a) \prod_{j=1}^{q} P(s_j | t = a)}{P(s_1, s_2, \dots \dots .., s_q)} \tag{1.13}$$

For another feature vector $s' = (s'_1, s'_2, \dots \dots .., s'_q)$ during testing data, a class T' is estimated using the given expression (Refer Eq. 1.14):

$$T' = argmax_c = 1,2, \dots \dots .., q \; P(t = a) \prod_{j=1}^{q} P(s'_j | t = a) \tag{1.14}$$

This classifier is easier to develop, understand, learns quickly, highly scalable and considers high dimensional features with small amount of training data because of independent assumption.

1.6.6.7 Random Forest (RF)

Random forest (RF) is a decision tree based ensemble classifier in which each tree grows due to randomization (Islam et al., 2019a). It is capable of processing large amounts of data due to decision tree with sufficient training speed. In random forest, the structure of each tree is binary and created in a top-down approach. It compute a response variable through various decision trees. For handwritten word recognition, most predicted class is imposed for multi-layered pixel object. Handwritten scanned word image, can be classified by sending it down to every tree and accruing the leaf ordination. Every tree is grown using a different subset by subsampling the training data (Kaur and Kumar, 2021b). These trees work as an ensemble and every individual tree makes the prediction of class. The model's prediction is carried out on the basis of voting as depicted in the Fig. 1.7. It acts an accurate and robust algorithm because the presence of many decision trees and takes the average of all the individual predictions.

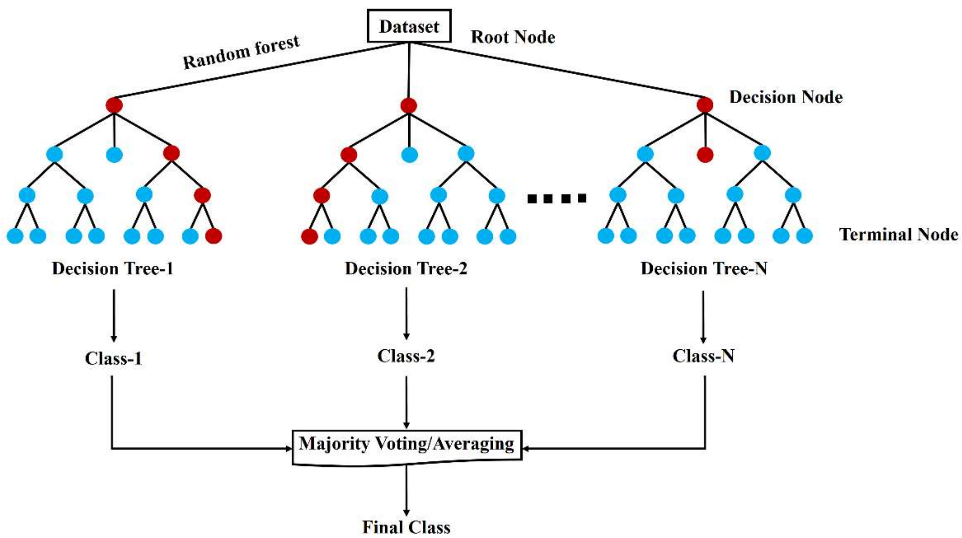

Figure 1.7: Random Forest (RF) classification

Binary test can be taken by applying either randomly or greedy approach at each node (Bosch et al., 2007). Mathematically, it can be expressed as (Refer Eq. 1.15):

$$\Delta G = -\sum_{i} \frac{|D_i|}{|D|} G(D_i) \tag{1.15}$$

Where, ΔG represents the information gain, D represents the dataset, D_i is the two subsets according to the given problem and $G(D_i)$ is the entropy with D_i the proportion of dataset in D belonging to a particular class.

1.6.6.8 Support Vector Machine (SVM)

A machine learning approach called Support Vector Machine (SVM) could be utilized for regression and classification (Rahim et al., 2013). SVM has the capability to learn high dimensional space with small training samples. Basically, it searches an optimal separating hyperplane and classifies the samples into classes as given in Fig. 1.8.

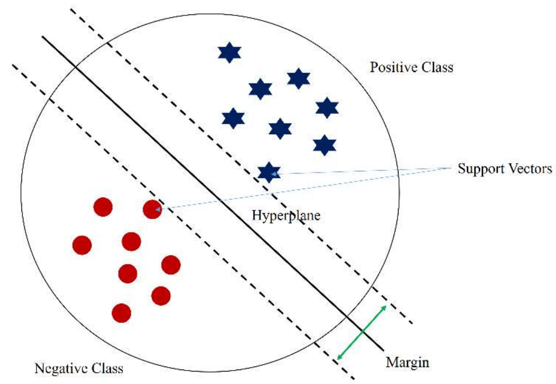

Figure 1.8: Support Vector Machine (SVM) classification

In general form, the separating boundary can be expressed using kernel trick as given in the following Eq. 1.16:

$$f(x) = sign\left(\sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b\right) \quad (1.16)$$

Where, N denotes training data samples, α_i $(0 \leq \alpha_i \leq C, i = 1, 2, \dots\dots\dots N)$ represents the parameter (learned from the data). The penalization misclassification cost parameter is given by C and is used in training data samples. x_i denotes the support vectors, b denotes a bias, y_i are the labels $(y_i \in \{-1,1\})$ and kernel function is denoted by $K(x_i, x)$. A Radial Basis Function (RBF) kernel can be expressed as follows (Refer Eq. 1.17):

$$K(x_i, x) = exp\left(-\frac{\|x_i - x\|^2}{2\sigma^2}\right) \quad (1.17)$$

The sigma (σ) is adjustable kernel parameter and plays important role to decide the performance of the RBF kernel. In this work, one-versus-one (OVO) approach has been explored due to its better performance for multi-class problems.

1.6.6.9 Extreme Gradient Boosting (XGBoost)

XGBoost, short for Extreme Gradient Boosting, is a highly effective machine learning algorithm widely utilized for diverse classification tasks, such as handwritten word recognition (Kaur and Kumar, 2021a). Employing an ensemble learning approach, XGBoost combines multiple weak learners, or decision trees, to form a robust predictive model. The algorithm employs gradient boosting to iteratively train decision trees, with each new tree aiming to rectify errors made by the ensemble up to that point (Ren et al., Li, 2017).

To leverage XGBoost for handwritten word recognition, it is essential to extract relevant features from the handwritten word images. These features encompass pixel intensities, texture descriptors, and shape-based representations, which capture the distinctive characteristics of the handwritten words.

1.6.7 Performance Evaluation

In order to analyze the performance of handwritten Devanagari word recognition system, an analytical study of different combinations of features and classifiers are carried out in terms of various performance evaluation metrics. The metrics considered for this experimental work include Recognition Accuracy (RA), Precision (PR), Recall (RL), F1-Score (FS), False Acceptance Rate (FAR), False Rejection Rate (FRR), Matthew's Correlation Coefficient (MCC) and Area Under the Curve (AUC).

These metrics can be obtained from confusion matrix. Various terms of confusion matrix include True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). These performance evaluation metrics are briefly outlined in the following subsections.

1.6.7.1 Recognition Accuracy (RA)

Recognition accuracy (RA) can be defined as the ratio of the number of correctly recognized test samples to the total number of input or test data samples (Ghosh et al., 2019).

It can be expressed as given in Eqs. 1.18 and 1.19:

$$RA = \frac{Number\ of\ correct\ predictions}{Number\ of\ input\ data\ samples} \times 100 \tag{1.18}$$

$$RA = \frac{(TP + TN)}{(TP + TN + FP + FN)} \times 100 \tag{1.19}$$

1.6.7.2 Precision (PR)

Precision indicates the proportion of positive identification that are right/correct. It can be defined as the ratio of correct positive results to the number of predicted positive results. Mathematically, precision may be defined as (Refer Eq. 1.20):

$$PR = \frac{TP}{(TP + FP)} \times 100 \tag{1.20}$$

1.6.7.3 Recall (RL)

Recall or sensitivity or True Positive Rate (TPR) indicates the proportion of actual positives that are identified rightly/correctly. It can be defined as the ratio of correct positive results and number of all relevant samples identified as positive. Recall value 1 means best sensitivity and 0 means worst sensitivity. Mathematically, it can be expressed as (Refer Eq. 1.21):

$$RL = \frac{TP}{(TP + \ FN)} \tag{1.21}$$

1.6.7.4 F1-Score (FS)

It indicates the harmonic average of the precision and recall. Its best value is 1 and worst is 0. For multiclass problems, it is calculated as weighted average of F1-Score of each class. It can be expressed as given in Eq. 1.22:

$$FS = 2 \times \left[\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}\right] = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \times 100 \tag{1.22}$$

1.6.7.5 False Acceptance Rate (FAR)

False Acceptance Rate (FAR) can be defined as the percentage of identification instances in which unauthorized instances are wrongly or incorrectly accepted. It can be expressed as the division of the number of false acceptances and the number of identification attempts. Mathematically, FAR can be given as given in Eq. 1.23:

$$FAR = \frac{FP}{(FP + TN)} \times 100 \tag{1.23}$$

1.6.7.6 False Rejection Rate (FRR)

False Acceptance Rate (FAR) can be defined as the percentage of identification instances in which unauthorized instances are wrongly rejected. It can be expressed as the division of the number of false rejections and the number of identification attempts.

Mathematically, FRR can be represented as given in Eq. 1.24:

$$FRR = \frac{FN}{(FN + TP)} \times 100 \tag{1.24}$$

1.6.7.7 Matthew's Correlation Coefficient (MCC)

Matthew's Correlation Coefficient (MCC) uses entire entries of confusion matrix. It is considered as better singular performance metric especially suited for imbalanced data training.

Mathematically, it can be expressed as given in Eq. 1.25 (Jurman et al., 2012):

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{1.25}$$

1.6.7.8 Area Under the Curve (AUC)

Area Under the Curve (AUC) represents the ability of a classifier to avoid wrong classification. Higher value of AUC indicates the better performance of the system for distinguishing between positive and negative classes.

For binary classification, it can be expressed as follows (Sokolova and Lapalme, 2009) (Refer Eq. 1.26):

$$AUC = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right) \tag{1.26}$$

1.7 OBJECTIVES OF THE PROPOSED WORK

The broad objectives of the thesis work are outlined as below:

- **I.** To generate a corpus of handwritten Devanagari words for experimental work.
- **II.** To explore existing features (structural and statistical) for offline handwritten Devanagari word recognition.
- **III.** To propose and implement innovative features for offline handwritten Devanagari word recognition.
- **IV.** To explore various classifiers for offline handwritten Devanagari word recognition.
- **V.** To calculate, compare and analyze recognition accuracy by combining various features and classifiers.

1.7.1 Assumptions

Following constraints have been considered for this work:

- **a)** The handwritten words are scanned at 300 dpi (dots per inch) resolution.
- **b)** Words are considered as noiseless and non-degraded.
- **c)** There is no requirement of skew detection and correction.
- **d)** Dataset/corpus does not contain any non-word items such as numbers, figures and tables etc.

1.8 MAJOR CONTRIBUTIONS AND ACHIEVEMENTS

Although, lot of efforts have been carried out for the development of handwritten Devanagari character recognition system in offline mode, however very few work exist for the development of the offline handwritten Devanagari word recognition framework. Therefore, the present work is an attempt in this regard/direction. Since there is the lack of publicly available benchmark/standard dataset in Devanagari (words) script, so we have gathered 48,000 samples of handwritten Devanagari words for experimental/research purposes. Also, we have proposed various approaches for the recognition of handwritten Devanagari words considering the combination of various features and classifiers which gives comparable recognition results with the available state-of-the-art techniques. Using combination of Gradient and structural-based features, we have obtained maximum recognition accuracy of 90.10% by exploring XGBoost classification approach.

To recognize handwritten Devanagari words, the recognition accuracies of 88.06% (with the combination of intersection & open-end points based features, elliptical based features and Arnold transform based directional features), 89.12% (with the combination/concatenation of uniform zoning, peak extent and Gabor filter-based features) and 94.53% (with the combination/ concatenation of uniform zoning, diagonal and centroid-based features) are obtained using majority voting classification, adaptive boosting and Gradient Boosted Decision Tree (GBDT) approaches, respectively. To the best of our knowledge, these considered combinations are the first of its type for the recognition of offline handwritten Devanagari words. Moreover, we have also proposed/developed an efficient approach for offline handwritten Devanagari word recognition using VGG16 as feature extractor (deep features) and XGBoost approach which results 95.00% of recognition accuracy.

1.9 ORGANIZATION OF THE THESIS

This chapter provides an insight on the classification of handwritten word recognition systems, its applications, pros and cons. The chapter gives an overview of the Devanagari script along with an elaboration of the various phases involved in offline handwritten word recognition system and discusses its application areas. Additionally, objectives of the proposed work, major contributions and achievements are provided. Chapter 2 focuses on the historical perspective of handwritten character and word recognition techniques/algorithms developed by various researchers that has been used in the various phases of the handwritten character/word recognition system. Research gaps identified through the literature review are also presented in this chapter. Moving on to Chapter 3, it examines various phases namely data collection, digitization, and pre-processing, which are essential for the offline handwritten word recognition system.

Chapter 4 presents an offline handwritten Devanagari word recognition system based on gradient and structural features along with their combinations. Three classification techniques namely Support Vector Machine (SVM), Naive Bayes (NB) and eXtreme

Gradient Boosting (XGBoost) are employed in this work. Chapter 5 analyzes recognition scheme for offline handwritten Devanagari words based on majority voting methodology. Three features namely intersection & open-end point-based features, elliptical-based features and Arnold transform-based features along with Support Vector Machine (SVM), Multi-Layer Perceptron (MLP) and Extreme Learning Machine (ELM) classifiers are used for classification. In Chapter 6, adaptive boosting approach is employed for offline handwritten Devanagari word recognition system considering three feature extraction techniques (uniform zoning-based features, peak extent-based features and Gabor filter-based features) and three classifiers (Support Vector Machine (SVM), Naive Bayes (NB) and Random Forest (RF)).

Chapter 7 introduces the Gradient Boosted Decision Tree (GBDT) approach to enhance the performance of offline handwritten Devanagari word recognition using uniform zoning-based features, diagonal-based features and centroid-based features. In Chapter 8, an efficient approach for offline handwritten Devanagari word recognition using deep features (VGG16) and XGBoost approach is proposed for offline handwritten Devanagari word recognition system considering three classifiers namely Naive Bayes (NB), XGBoost and Random Forest (RF)). Finally, Chapter 9 presents brief contribution of the work, major conclusions and suggests future scope of the work for further research.