Computer Systems Science & Engineering
DOI:10.32604/csse.2022.024059
Article

Tech Science Press

Support Vector Machine Based Handwritten Hindi Character Recognition and Summarization

Sunil Dhankhar[1,*], Mukesh Kumar Gupta[1], Fida Hussain Memon[2,3], Surbhi Bhatia[4], Pankaj Dadheech[1] and Arwa Mashat[5]

[1]Department of Computer Science & Engineering, Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur, Rajasthan, 302017, India
[2]Department of Electrical Engineering, Sukkur IBA University, Pakistan
[3]Department of Mechatronics Engineering, Jeju National University, Jeju, 63243, Korea
[4]Department of Information Systems, College of Computer Science and Information Technology, King Faisal University, Saudi Arabia
[5]Faculty of Computing & Information Technology, King Abdulaziz University, Rabigh, 21911, Saudi Arabia
*Corresponding Author: Sunil Dhankhar. Email: sunil@skit.ac.in
Received: 02 October 2021; Accepted: 08 November 2021

Abstract: In today's digital era, the text may be in form of images. This research aims to deal with the problem by recognizing such text and utilizing the support vector machine (SVM). A lot of work has been done on the English language for handwritten character recognition but very less work on the under-resourced Hindi language. A method is developed for identifying Hindi language characters that use morphology, edge detection, histograms of oriented gradients (HOG), and SVM classes for summary creation. SVM rank employs the summary to extract essential phrases based on paragraph position, phrase position, numerical data, inverted comma, sentence length, and keywords features. The primary goal of the SVM optimization function is to reduce the number of features by eliminating unnecessary and redundant features. The second goal is to maintain or improve the classification system's performance. The experiment included news articles from various genres, such as Bollywood, politics, and sports. The proposed method's accuracy for Hindi character recognition is 96.97%, which is good compared with baseline approaches, and system-generated summaries are compared to human summaries. The evaluated results show a precision of 72% at a compression ratio of 50% and a precision of 60% at a compression ratio of 25%, in comparison to state-of-the-art methods, this is a decent result.

Keywords: Support vector machine (SVM); optimization; precision; Hindi character recognition; optical character recognition (OCR); automatic summarization and compression ratio

1 Introduction

Automatic writing is a sophisticated computer technique that attempts to digitally encrypt printed or handwritten information to be read by a machine. Written recognition specifically includes all duties

about the processing of bulk paper documents. Thus, a vast base of applications is created, in-including automated administrative file management, automatic mail sorting, read amounts and bank checks, mailing address processing, form processing, keyboard-free interfaces, and textual analyses, read legacy documents, library archiving indexing, and data search. Automating one of these instances is a challenging problem because the writer's habits, styles, and formats are highly unpredictable. The device must first understand the subject and employ a robust mathematical formalism. In the last decade, automated reading has dramatically improved. Many research activities have resulted in a wide range of methods, including the performance of computers and the current acquisition system, combined with modern statistical approaches such as cache models, vehicle support machines, and neural networks.

1.1 Basic of Character Recognition

Printing and handwriting are two areas of study. The identification of the handwritten input method has two application areas. If the input modalities are dynamic or online, recognition is addressed in real-time. Offline recognition refers to delayed time recognition [1]. Offline recognition occurs when text is scanned and saved as an image on paper. This image comprises binary (black and white) or integer pixels (grayscale). Offline identification is addressed by identifying handwritten features and identifying words.

The development of a handwriting recognition system is divided into several stages. The steps outlined by Duda et al. [2] are depicted in Fig. 1. The system receives an input or a sensor signal (camera, scanner, or tape recorder). Following that, several methods for these photos and data are developed. The goal is to eliminate the phenomena that contribute to system efficiency degradation, reduce quantification noise (binarization), and keep the related components in the picture link. This step generates a vector function for the classification module's heading for each image.

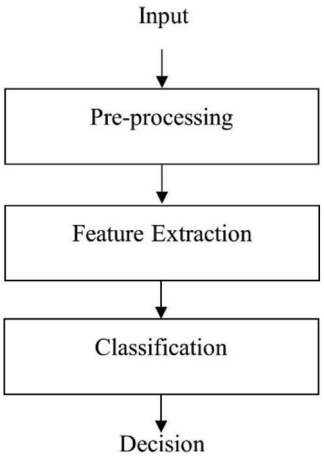


Figure 1: The steps of character recognition

1.2 Hindi Character Recognition

The recognition of handwritten Hindi characters is proposed in this study. Devanagari is the Hindi language's mother script. In Hindi, there are 13 vowels, 36 consonants, and 14 modifiers, as shown in Fig. 2 [3]. In Hindi, vowels are referred to as "Swar", while consonants are "Vyanjan". In general, handwriting recognition aims to create a system as close to human reading ability as possible. However, the purpose of handwriting recognition is to extract a more concise description of an anonymous form (word, letters, numbers) and base a decision on it. This decision is typically made by comparing an unknown form to a set of references stored in memory and described similarly. The concerns come from

a qualified previous stage of learning. This stage is critical in any handwriting recognition system. In other words, it is a transition from the observable space to the decision space of class membership [4].

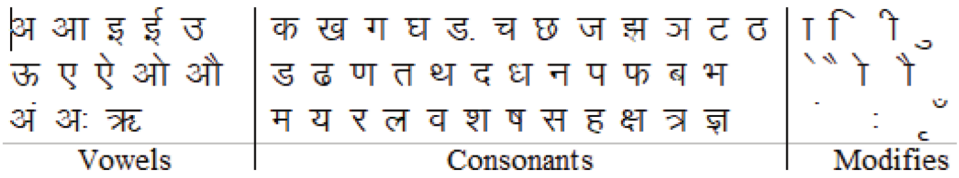


Figure 2: Hindi vowels, consonants, and modifies

1.3 Hindi Text Summarization

Automatic Text Summary (ATS) condenses large amounts of information into a smaller text, including essential knowledge, subjects, and other original text components. The machine has a text and returns a summary of the original text. Text synthesis may be helpful in everyday situations. Film reviews, news headlines, abstract explanations of technical work, and book reviews are just a few examples. ATS is an important step in information management. It is concerned with the selection of the essential sections of the text.

Summaries are classified into two types: extractive summaries and abstractive summaries [5,6]. When all terms from the original text are removed, extractive summaries are produced. The statistical and linguistic characteristics of sentences influence their value. An abstract overview identifies the key ideas in a text and presents them to the reader directly. Language techniques are used to analyze and comprehend a text, as well as to find new ideas and words to explain them better and to create short new messages based on the preliminary information from the original text [7]. While there are numerous methods for summarizing important information, some complicated problems are available in other languages worldwide, such as English, Swedish. There are approximately 120 major languages in India, with an additional 1500. Over the last few years, many academics have worked in Indian and other languages, but little research has been done on the Hindi language. While Hindi is India's most widely spoken language, many neighbors lack an effective Hindi text synchronization system. As a result, we choose the Hindi language.

2 Literature Review

One of the most well-known ideas for automated pattern recognition is the identification of the optical character. The process begins with reading, evaluates the meaning of a scanned image of character evolution, and then translates the image into computer text content. Post offices frequently use this process to read addresses and names automatically and check the amount and number of checks in wrappers and banks. Gaur et al. [7] presented a three-step technique for identifying the Hindi characters. The first stage of pre-processing includes image binarization and character separation. The Extraction of functionalities occurs in the second stage, which is accomplished by grouping k-means and creating the function vector. Classification using SVM is the final step.

Handwritten character recognition (HCR) and optical character recognition (OCR) are currently receiving much attention. The OCR framework is appropriate for multi-decision inquiries, public address definition, and other applications. In contrast, HCR is more commonly used than OCR, which is helpful in a variety of processing frameworks. In the future, the character recognition framework may aid in digitization and create a paperless sector. The application for optical character identification is reorganized using Artificial Neural Networks (ANNs), which have excellent recognition accuracy and high performance [8]. The backpropagation method is used to train neural networks [9]. S. Mehfuz et al. [10] provide an extensive assessment of current studies in handwritten character recognition considering the soft-computer approach over the last decade. A handwritten character recognition method was proposed

with a high accuracy level (90%) [11]. Mandal et al. [12] improve the performance of previous methods by reducing the image matrix's irrelevance; it is compressed into a smaller matrix for a specific purpose. Each component of the picture matrix is mapped to a different pattern. Most of a way selects a specific character. The ability of the ANN to recognize extensive collections of optical imagery has been demonstrated using a propelled categorization system based on probabilistic neural networks [13].

Patil et al. [5] presented an offline recognition method for Hindi characters based on ANN algorithms. The first step is to divide characters into lines, words, and characters, followed by a feed-forward neural network technique for chat character identification. Singh et al. [14] an overview of OCR use is provided, and a study is conducted for three major applications: captcha, institutional repository, and optical music character identification. Su et al. [15] proposed Offline handwritten letter alphabet recognition using a multilayer feed to a neural network system with fifty data sets, each consisting of 26 letters from different people, are used to prepare the neural system, and 570 different texts are used to test characters. A study was also conducted to determine the number of concealed nodes required to conduct a backpropagation network performance evaluation [16].

Syed et al. [17] presented a hidden layer multilayer perceptron to demonstrate character recognition. The character is identified by examining its forms and comparing its features. A mouse-equipped framework for the perception of handwritten letters or images was proposed [18]. [19,20] presented a comparison of opinion summarization techniques and proposed an unsupervised algorithmic approach for opinion summarization that does not require annotated training data, adequately classifies the input text and achieves 96% accuracy. [21] classify the raw text using the danger theory of the Artificial Immune System (AIS). They use krill herd optimization (KHO) for feature selection, and various optimization functions (Quing function, Sumsquare function, Levy function) are applied for enhancing its performance. The frame provides a method for first preparing the input characters, and then there is a good option for feeding the previously created patterns or pictures to identify them. Text summarization extracts data from extensive texts [22]. An unstructured text is analyzed, and a network of weighted nodes is created. Starting with pre-processing, the technique proceeds to text processing via forwarding and backward propagation. This method saves time and money by searching the most relevant parts of the corpus. The growth of internet data has made automatic text summarization work more relevant than ever. That data may be used for a variety of purposes. It uses fuzzy logic, multi-features, and a Genetic algorithm to summarize news [23]. Because news material often includes elements like time, place, and characters, these elements can be extracted as keywords. Sentence characteristics determine a sentence's importance and test on the DUC2002 dataset.

3 Background and Significance

3.1 Recognition of Text or Analysis of Documents

In the initial step, a structured text with a few sentences or words should be recognized. The search is carried out with the comprehensive acknowledgment of the terms in the sentences followed by a transformation into the character of each word [24]. The second phase (document analysis) includes adequality organized information, which requires a knowledge of Material layout and typography. The more straightforward pre-processing method is thus not used when a particular step is taken into regional location, Graphics and photographic areas separation, Semantic tagging of model textual regions, and determining reading order and document structure [25].

3.2 Recognition of Print or Manuscript

Most printed characters are aligned horizontally and vertically, making reading more straightforward [26,27]. The letter form is based on the calligraphic style are usually linked, and the visuals connecting

them are irregularly proportionate. This typically involves the use of specific assignment techniques and sufficient information to enable interpretation [28]. Mono-fonts, multi-fonts, or Omni-fonts may be included in the printing. Mono-fonts is a system if a single font can only be recognized. A multi-font system is considered when a collection of typefaces previously taught recognizes several types. Furthermore, any font may be identified in an Omni-font system, generally without learning [29,30]. However, this is almost difficult because hundreds of kinds exist, some of which are humanely unreadable. The global approach considers the word an entity and defines it regardless of characters. The advantage of this technique is that it preserves the character in its context that makes it possible to model change of the writing and its potential degradation more effectively. This method, however, undermines the memory capacity, computer time, and complexity of the treatment, linear to the Lexicon, which limits vocabulary [31,32].

3.3 General Organization of a Recognition System

Recognition systems frequently use the following stages:

3.3.1 Straightening of the Writing

One of the problems encountered in OCR is the slant of the lines of the text, which introduces difficulties for segmentation. The tilt can come from the input if the document has been placed at an angle or be intrinsic to the text. It should be straightened to find the structure of horizontal lines of a text image. If α is the angle of inclination, to straighten the image, an isometric rotation of angle α is carried out thanks to the following linear transformation in Eq. (1) [33]:

$$X' = x\cos\alpha + y\sin\alpha, \quad Y' = y\cos\alpha + x\sin\alpha \tag{1}$$

3.3.2 Smoothing

The image of the characters may be tainted with artifacts due to acquisition and the quality of the document, leading to either no points or overhead. Smoothing techniques solve these problems by local operations called clogging and cleaning operations [34].

3.3.3 Standardization

After normalization of the size, the images of all the characters are found defined in a matrix of the same size to facilitate the subsequent treatments. This operation usually introduces slight deformations on the images. However, certain characteristic features such as the stem in the characters can be eliminated because of normalization, leading to confusion between certain characters [33].

3.3.4 Skeletonization

The purpose of this technique is to simplify the image of the character into an image that is easier to deal with by reducing it to the character line. Skeletonization algorithms are based on iterative methods. The process is done in successive passes to determine if a particular pixel is essential to keep it or not in the plot [33].

3.3.5 Segmentation Phase

In this phase, the different logical components of an image are extracted. From the recorded image, text and graphic blocks are separated first and removed from a text block from which words and letters are found [29].

3.4 Feature Extraction

This is one of OCR's most sensitive and essential stages. A character recognition starts by analyzing its shape and extracting its distinctive characteristics, which are used to identify it. Feature types may be categorized into four major groups: structural characteristics, statistical features, global changes, overlaps, and correlations of the model [29,35].

3.4.1 Structural Features

The structural characteristics define a form that provides world and local attributes in terms of topology and geometry. These characteristics may be features and handled in various orientations and sizes, the final points, points of intersection, buckles, the location of diacritical points with relation to the baseline, the vows and zigzags, the character's height and breadth, the form category. Several additional properties may be obtained, depending on whether the curve, line, or contour segment is extracted [35].

3.4.2 Statistical Characteristics

The features utilized for recognizing the texture: zoning, location (Loc_i) features, and moments [35]. Zoning consists of superimposing a grid $n * m$ on the image of the character and for each of the resulting regions, calculating the average or the percentage of points in grayscale, thus giving a vector of size $n * m$ of characteristics. Loc_i determine the number of segments of white and black along a vertical line crossing the shape and the length [29]. The moments of a form are invariant concerning its center of gravity and maybe invariant about rotation [31].

3.4.3 Global Transformations

The transformation consists of transforming the image into a more abstract representation to decrease the character dimension and preserve as much information as possible about the shape to be identified. The skeleton or touch of a character in a string of direction codes is one of the most straightforward transformations [30].

3.4.4 Template Matching and Correlation

The 'template matching' technique applied to a binary picture (grayscale or skeleton) involves comparing the form image as a vector of features to a pixel-by-pixel model in recognition. A similarity measure is determined [35].

3.4.5 Classification Phase

The categorization of the OCR system consists of two tasks: learning, recognition and decision-making. At this point, the features of the previous stage are utilized to identify and assign a text fragment to a reference model [35]. In the learning phase, the system is taught about the critical characteristics of the language used and is organized into reference models. The goal is to teach the system as many samples as many forms of writing; however, that is difficult due to the enormous diversity of writing that would lead to combined models of representation explosion [27,30]. Learning consists of two concepts: education and adaptability. Training teaches the system character description and adaptation to enhance the system performance by drawing on past experiences [30]. The learning methods vary from recognizing printed to handwritten characters to identifying mono-font or multi-font texts. Two kinds of learning methods exist in general: supervised and unattended. Learning is supervised if the supervisor is a teacher. It is done by introducing many reference samples during a preliminary recognition phase. Recognition and Decision Making is the last recognition phase. This phase finds the closest description parameter models for the character processed. Recognition may succeed if the answer is unique. It may lead to confusion if numerous solutions are found. Finally, if no model conforms to its description, it may reject the form. A probability measure may accompany the choice in the first two instances, also termed the score or recognition rate [30].

3.4.6 Post-processing Phase

The purpose of the post-processing technique is to improve words. This phase is often performed as a collection of character frequency tools by string, lexicons, and other contextual information. The classification may lead to many potential options, so post-processing seeks to use a higher degree of

information to choose the answer. Post-processing additionally verifies whether the response is accurate, depending on other information which the classifier cannot access [30].

4 Proposed Methodology

In the proposed methodology describes all the steps which are used in the implementation.

4.1 System Model

The flow diagram of the proposed character recognition system is shown in Fig. 3:

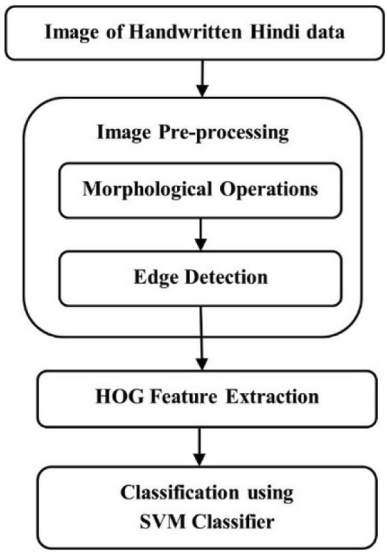


Figure 3: Flow diagram of proposed work

4.2 Data Collection

A scanned image of the alphabet and numeric data in joint photographic experts group (JPEG) format is loaded to the system. The system we develop here can identify the character. So, the user must select the character of its own choice for recognition input. Once the input character is selected, several processes are applied to it.

4.3 Pre-processing

The purpose of pre-processing is to assist a form (vowels, consonants, and modifications) characterization. The purification of images is essentially designed to remove leftover noise from the binarization. The operation may reduce the quantity of information to be processed to transfer the line thickness to a single pixel or by monitoring or extracting upper, lower, or interior contours. Standardization is to correct the slope of a word or correct the inclination of the letters in a comment to facilitate segmentation. Segmentation is a pre-processing step. It aims to identify and extract the information to be recognized as accurately as possible. The segmentation is done utilizing edge detection and morphological procedures in this study.

4.4 Method for Extraction Based Summarization Techniques

There are many methods for generating extractive summaries that may be appended to the appropriate phrase's summary. Here, we are describing the only statistical and linguistic method. In the statistical method,

the produced text summary is based on the statistical distribution of specific characteristics and is carried out without comprehending the whole content. Classification techniques classify phrases included in the summary based on sentence position, sentence length, or word occurrences in the text. This technique retrieves phrases from the source texts, regardless of the semantics of the words [1]. In contrast, a linguistic method must know about the language so that the computer can evaluate the sentences and then pick which sentence to choose. It detects word relationships via speech tagging, grammatical analysis, thesaurus use, and the extraction of significant phrases in the text. Cue words, titles, or nouns and verbs in phrases may be parameters [3].

4.4.1 Morphological Operations

Mathematical morphology uses the structural element [4] to process images in the form of specific forms already chosen, typically less than the image, that function as a result operator on an image. The shape of the structural element, size, and orientation are selected according to previous knowledge by the relevant geometrical structures existing in the picture and the goal of the morphological operation conducted [4].

Dilation: Dilation (Eq. (2)), also referred to as expansion, filling or growth has a thickening impact on the border. This method is used to stretch object contours and theoretically define binary dilation as discontinuous lines of those produced using certain filters:

$$A \oplus B = c \in E^N \mid C = a + b \; for \; all \; a \in A \; and \; b \in B^{\Sigma} \tag{2}$$

Erosion: Erosion (Eq. (3)) is a double expansion feature but not the other way around. If erosion is done and the generated picture is not the same as the actual image, erosion is mathematically defined as:

$$A \Theta B = x \in E^N \mid x + b \in A \; for \; all \; b \in B^{\Sigma} \tag{3}$$

Fig. 4 gives an example of dilation and erosion operations with the given structuring element.

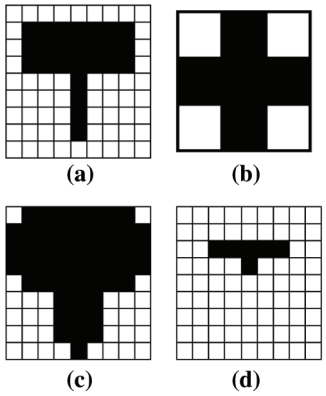


Figure 4: Illustration of morphological operations [4]. (a) Original image. (b) Structuring element called simple cross. (c) Dilation of (a) by (b). (d) Erosion of (a) by (b)

4.4.2 Edge Detection

It significantly reduces the amount of data in a picture while preserving structural features for subsequent image processing. Several edge detection methods exist, like canny edge detection (CED) [36]. It is very ancient but one of the standard edge detection techniques and is still in use in research. The CED aimed to create an algorithm that satisfies.

- **Detection:** Maximize the chance of identifying genuine edge points while minimizing the likelihood of erroneous detection of non-edge issues.
- **Localization:** The identified edges should be as near to the actual edges as possible.
- **Number of Responses:** A true edge should not lead to more than one edge.

4.5 Feature Extraction

The purpose of feature extraction in recognition is to express the feature in the numerical or symbolic form called encoding. Depending on the case, the values of these features can be confirmed, integer, or binary. The Histograms of Oriented Gradient (HOG) is a contour description. Every window with the local presentation and amplitudes distribution may be seen in this scenario. This distribution has a histogram to define. This splits the location window into a cell setting that includes the gradient adaptation across the cell pixels at each orientation interval. The picture gradients may be computed using a Sobel operator at each location (x, y) (see Eq. (4)),

$$G(x,\ y) = \sqrt{G_x(x,\ y)^2 + G_y(x,\ y)^2} \tag{4}$$

where G_x and G_y The gradient of the axes of x and y is thus present at this location (see in Eq. (5)),

$$\theta(x,\ y) = arctan\frac{G_y(x,\ y)}{G_x(x,\ y)} \tag{5}$$

The contours are then divided into K intervals (bins). The interval value of kth is determined by Eq. (6),

$$\psi_k(x,\ y) = \begin{cases} G(x,\ y) & si\ \theta(x,\ y) \in bin_k \\ 0 & if\ not \end{cases} \tag{6}$$

The k(x, y) computation of an area R (see Eq. (7)) of the picture may be done effectively utilizing the integral image concept.

$$SAT_k(R) = \sum_{(x,y)\in R} \psi_k(x,\ y) \tag{7}$$

4.6 Classification

The categorization creates a decision-making rule which converts characteristics describing the forms into class membership. Before a decision model is incorporated into a handwriting recognition system, there are two steps beforehand: the learning step and the test step. There are two kinds of learning: supervised and unsupervised [37]. In the case of supervised learning, the learning phase is analyze the similarities between labeled elements of the same class and other classes so that the optimal score in the representation space is obtained. In the event of unsupervised learning, a great number of unmarked forms are given to the recognition system. The categorization phase automatically identifies the states of the same class. The test phase assesses the classifier's performance for learning. The iterations are designed to extract features that are considered helpful for issue identification. When the required system performance is not reached, a new family of features will be found again, or new features extracted are combined. The calculation of this performance is the result of the SVM classifier used for the classification of the Hindi character recognition system, which is described in the following heading.

4.6.1 Support Vector Machine (SVM)

Consider the training set $\{x_1, y_1\}, \ldots, \{x_\ell, y_\ell\}$, where $x \in X$ and $y \in \{-1, 1\}$, where ℓ is the number of observations and X is a distribution in space $\Re^n$. In the classification problem, the goal is to find an efficient

method to construct the optimal separator hyperplane, i.e., with the greatest margin. To do this, one must find the vector w and the constant b, which minimize the norm $|w|^2 = w^T w$ (since it is inversely proportional to the margin), under the constraints:

$$w^T x_i + b \geq 1, \qquad if\ y_i = 1 \tag{8}$$

$$w^T x_i + b \leq -1, \qquad if\ y_i = -1 \tag{9}$$

Because one can accept some errors, one relaxes the constraints Eqs. (9) & (10) and introduces an additional cost related to this relaxation so that one arrives at the quadratic problem, QP, following Eq. (10)

$$\begin{aligned} &Minimize && \frac{1}{2}(w^T w) + C\left[\sum_{i=1}^{\ell} \xi_i\right] \\ &Under\ \ the\ \ constraint && y_i(w_i^T x + b) \geq 1 - \xi_i, \quad \text{Where } 0 \leq \alpha_i \leq Ci = 1, \ \ldots, \ \ell \end{aligned} \tag{10}$$

The Eq. (10) can be solved in the primal space (the space of parameters w and b). One solves the QP in the dual space, Eq. (11) (the Lagrange multiplier space) for two main reasons: 1) The constraints Eqs. (9) and (10) are replaced by the associated Lagrange multipliers, and 2) We obtain a formulation of the problem where the training data appear as an internal product between vectors, which can then be replaced by kernel functions, then construct the hyperplane in the feature space and obtain functions Non-linear in the input space (see Eq. (11))

$$\begin{aligned} &\text{Maximize } L_D(\alpha) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j (x_i^T x_j) \\ &\text{Under the constraints } \sum_{i=1}^{\ell} y_i \alpha_i = 0, \quad \text{Where } 0 \leq \alpha_i \leq Ci = 1, \ \ldots, \ \ell \end{aligned} \tag{11}$$

where α_i is the Lagrange multiplier associated with constraints. Parameter C controls the level of error in the classification. The SVM evaluation function is defined in Eq. (12)

$$f(x) = \sum_{i=1}^{\ell} \alpha_i y_i k(x_i, \ x) + b \tag{12}$$

The examples x_i associated with multipliers α_i larger than zero correspond to the support vectors and have a significant contribution to Eq. (12). The constant b represents the threshold of the hyperplane learned in the characteristic space.

4.6.2 Pre-processing

Pre-processing involves the following:

- **Segmentation of Sentence:** Each input document d_i consists of several sentences $S_1, S_2, \ldots, S_n$, and the number of phrases in the document, each S_i representing the phrase in the text.
- **Tokenization:** The phrases in each sentence are referred to as $t_1, t_2, \ldots, t_m$, where all t_j is indicated in sentence S_i.
- **Stop word removal:** Stop words are the most frequent terms and must be filtered out
- **Stemming**: It is a method to reduce inflected words to their tongues using Indic stemmer.

4.6.3 Sentence Encoder After Pre-processing

The next step is to translate each phrase with actual numbers into a fixed-length vector. An image of the D_i document's space vector is a vector image of each D_i word x. The suggested approach utilizes several word integration devices such as term frequency-inverse document frequency (TF-IDF), word-to-vector (Word2Vec), and Smooth Inverse Frequency (SIF) for vectorizing the sentence. TF-IDF method analyses how frequently the term (TF) is discovered in a particular document and multiplies it by the value that a word is expected in the entire collection of documents (IDF) [1]. In Word2Vec, words are mapped to vectors using the model of Continual Skip-Gram (CBOW) or Skip-Gram. In SIF, the phrase is vectorized using FastText, a word model learned in common crawl and Wikipedia. Kolman et al. [4] proposed an alternative method, termed a SIF described in Algorithm 1.

4.7 Pseudo Code

In this section, we discuss three algorithms: Algorithm 1 identifies the edges of an image using the pixel intensity then generates the text, finally, the generated text is matched with the Hindi language characters.

Algorithm 1: Edge image to text generation and classification of text

Image generate_Edge_text_TDIF Image

$l \leftarrow 0$, $k \leftarrow 0$, *left* $\leftarrow 0$, *upper* $\leftarrow 0$, *rightUpper* $\leftarrow 0$

for all $pixel_{l,k} \in grayImg$ ***do***

if $(0 < l < L - 1)$ *and* $(0 < k < K)$ ***then***

$left \leftarrow |pixel_{l,k} - pixel_{l-1,k}|$

$upper \leftarrow |pixel_{l,k} - pixel_{l,k-1}|$

$rightUpper \leftarrow |pixel_{l,k} - pixel_{l+1,k-1}|$

$edgeImg_{l,k} \leftarrow$ ***max***(*left*, *upper*, *rightUpper*)

else

$edgeImg_{l,k} \leftarrow 0$

end if

end for

edgeImg $\leftarrow$ *sharpen*(*edgeImg*)

return(*edgeImg*)

Algorithm 2 is used for detected edges should be as close as feasible to the real edges. After successful identification and generation of text, we apply Algorithm 3 for text summarization that first finds the text features then summarizes the input text according to identified features.

Algorithm 2: Text image localization

text Region [ ] *detect Text Regions*(*Image edgeImg*)

Integer[ ] $H \leftarrow$ *calculate Line Histogram*(*edgeImg*)

text Regions [ ] *TC* $\leftarrow$ *determine Y coordinate* (*H*)

TC $\leftarrow$ *determine L coordinate* (*edgeImg*, *TC*)

return(*TC*)

Algorithm 3: Text summarization

image segment text features (*HoG*, *edgeImg*[ ] *TC*)

Comment: *HoG − Histograms of Oriented Gradients*

Image reducedImg ← *erase*(*TC*, *Img*)

Image binaryImg ← *binarize* (*binaryImage*)

Image gapImg ← *fillsGaps*(*binaryImg*)

TC ← *refine coordinates* (*HoG*, *edgeImg*, *gapImg*, *TC*

Image textImg ← *extractImage*(*grayImg*, *TC*)

textImg ← *enhanceContrast*(*textImg*)

return(*textImg*)

5 Result and Analysis

5.1 Dataset

Dataset is the collection of organized data that is used in an experiment. The research of this presented work focuses on the ambiguity of the Hindi language, so Hindi ambiguous words are our primary source of Hindi document. The dataset for the experiment is collected from the technology development for Indian languages (TDIL) and due to the unavailability of a large and open data set, a small testing data set is also created manually with the help of various standard online essays, news, or history. The characteristics of the dataset are stated in see Tab. 1.

Table 1: Characteristics of the dataset

Categories	Politics, Science, Sports, History
The average number of sentences per context	30–40
The average number of words per context	400–500

5.2 Evaluation Parameter and Result

Evaluation of a result is necessary to check its usefulness and relevance. Accuracy is measured as the number of correct senses obtained out of all senses when evaluating generated disambiguation. Generated meaning of an ambiguous word is also evaluated manually by human experts, but the problem is that every individual who performs evaluation has a different idea about the meaning of the word. One more problem is that it is time-draining because it is complicated to go through the entire document. So, Accuracy is calculated as the number of correct senses obtained for every sentence in the context by the total number of appearances of the word. The accuracy obtained by the proposed method is compared with ensemble subspace discriminant (ESD), Quadratic SVM, k-nearest neighbor (k-NN), weighted k-NN, and bagged trees. The comparison is shown in Tab. 2.

Table 2: Accuracy and precision score obtained by different classifiers

S. No.	Classifiers	Accuracy (%)	Precision (%) at compression rate	
			25%	50%
1	ESD	94.7	57	70.8
2	Quadratic SVM	96.6	59	71.54
3	k-NN	93	59.5	67
4	Weighted k-NN	93.1	59.3	68.5
5	Bagged Trees	93.2	59.4	69
6	**HOG with SVM (proposed)**	**96.97**	**60**	**72**

We found that our proposed method gives good accuracy as compared to other methods. The graphical comparison is shown in Fig. 5.

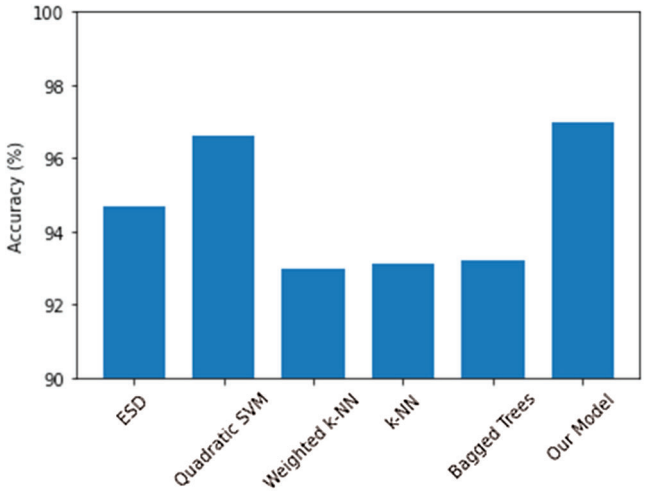
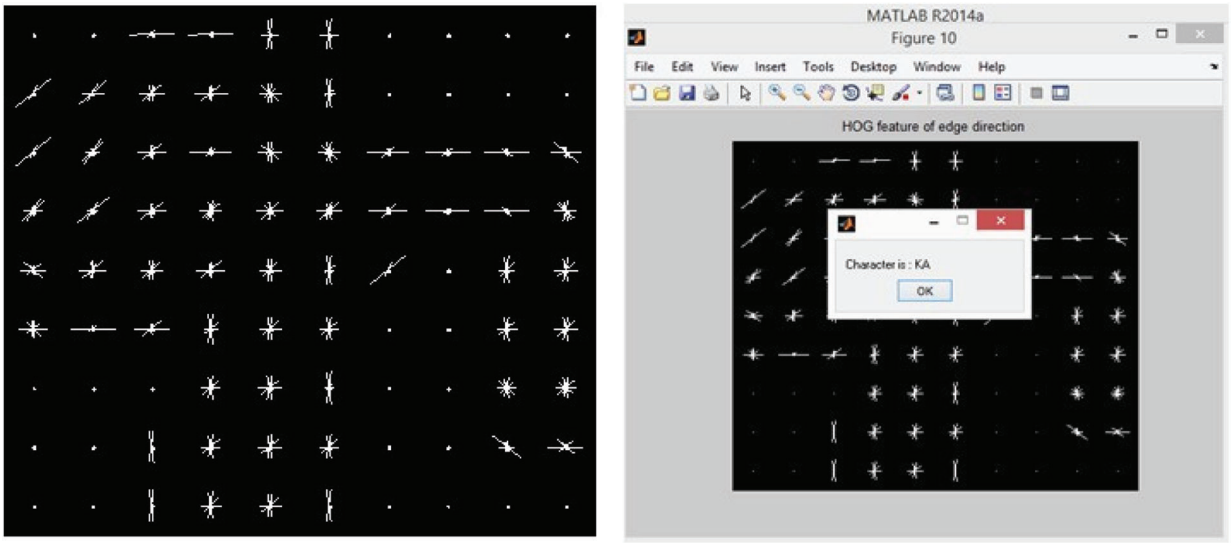


Figure 5: Accuracy graph

5.3 Example

The sample example for word sense disambiguation is shown in Fig. 6a. The output is shown in Fig. 6b. According to the Hindi wordnet, the input text file has checked for various nouns having more than one meaning. The experiments are performed for multiple domains like sports, literature, news, or history.



(a) (b)

Figure 6: Predication of correct Hindi character. (a) HOG feature of edge detection. (b) Recognized character

6 Conclusion and Future Scope

The work in this article addresses the processes needed to create a system for the identification of Hindi characters. For each of these steps, we attempt to offer a technique of optimization to choose characteristics of the appropriate detection system: pre-processing, extraction of features, and classification. Thus, the selection of relevant and non-redundant systems is made from the extraction phase of the feature. This option reduces classifier (SVM) inputs while increasing or maintaining the recognition rate. The examination of the implementation of the morphological operation and the edge detection in selecting HOG features is the main contribution of this article. The recognition accuracy of the proposed model is 96.97% and the precision score is 72% and 60% at the compression rate of 50% and 25% respectively. The accuracy of recognition of the SVM method presented here may be further enhanced as a future aspect. The method presented cannot be used to recognize a word. In the future, therefore, a Hindi word may be recognized.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] G. Katiyar and S. Mehfuz, "MLPNN based handwritten character recognition using combined feature extraction," in *Int. Conf. on Computing, Communication & Automation*, IEEE, Greater Noida, India, 2015, pp. 1155–1159.

[2] R. O. Duda and P. E. Hart, "Linear algebra, Lagrange optimization Probability theory," in *Pattern Classification*, 1st ed., USA: John Wiley & Sons, pp. 8–24, 2006.

[3] R. Jayadevan, S. R. Kolhe, P. M. Patil and U. Pal, "Offline recognition of Devanagari script: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 6, pp. 782–796, 2011.

[4] E. Kolman and M. Margaliot, "A new approach to knowledge-based design of recurrent neural networks," *IEEE Transactions on Neural Networks*, vol. 19, no. 8, pp. 1389–1401, 2008.

[5] M. Patil, B. Mitrakshi and V. Narawade, "Recognition of handwritten Devanagari characters through segmentation and artificial neural networks," *International Journal of Engineering Research and Technology (IJERT)*, vol. 1, no. 6, pp. 1–5, 2012.

[6] T. Dash and T. Nayak, "English character recognition using artificial neural network," *arXiv preprint arXiv:1306.4621*, 2013.

[7] A. Gaur and S. Yadav, "Handwritten Hindi character recognition using k-means clustering and SVM," in *Proc. 4th Int. Symp. on Emerging Trends and Technologies in Libraries and Information Services*, IEEE, Noida, India, 2015, pp. 65–70.

[8] S. Barve, "Artificial neural network based on optical character recognition," *International Journal of Engineering Research & Technology (IJERT)*," vol. 1, no. 4, pp. 1–5, 2012.

[9] M. Alojail and S. Bhatia, "A novel technique for behavioral analytics using ensemble learning algorithms in Ecommerce," *IEEE Access*, vol. 8, pp. 150072–150080, 2020.

[10] S. Mehfuz and G. Katiyar, "Intelligent systems for offline handwritten character recognition: A review," *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 4, pp. 538–544, 2012.

[11] D. K. Patel, T. Som, S. K. Yadav and M. K. Singh, "Handwritten character recognition using multiresolution technique and Euclidean distance metric," *Journal of Signal and Information Processing*, vol. 3, no. 2, pp. 208–214, 2012.

[12] R. K. Mandal and N. R. Manna, "Handwritten English character recognition using column-wise segmentation of image matrix (csim)," *WSEAS Transactions on Computers*, vol. 11, no. 5, pp. 148–158, 2012.

[13] P. J. Simha and K. V. Suraj, "Unicode optical character recognition and translation using artificial neural network," in *Int. Conf. on Software Technology and Computer Engineering (STACE-2012)*, Vijayawada, India, pp. 27–31, 2012.

[14] A. Singh, K. Bacchuwar and A. Bhasin, "A survey of OCR applications," *International Journal of Machine Learning and Computing*, vol. 2, no. 3, pp. 300–314, 2012.

[15] M. Su, C. Wu and H. Cheng, "A two-stage transformer-based approach for variable-length abstractive summarization," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 2, no. 3, pp. 314–318, 2012.

[16] J. Cheng, F. Zhang and X. Guo, "A syntax-augmented and headline-aware neural text summarization method," *IEEE Access*, vol. 8, pp. 218360–218371, 2020.

[17] A. A. Syed, F. L. Gaol and T. Matsuo, "A survey of the state-of-the-art models in neural abstractive text summarization," *IEEE Access*, vol. 9, pp. 13248–13265, 2021.

[18] M. Y. Saeed, M. Awais, R. Talib and M. Younas, "Unstructured text documents summarization with multi-stage clustering," *IEEE Access*, vol. 8, pp. 212838–212854, 2020.

[19] S. Bhatia, "A comparative study of opinion summarization techniques," *IEEE Transactions on Computational Social Systems*," vol. 8, no. 1, pp. 110–117, 2020.

[20] S. Bhatia, M. Sharma and K. K. Bhatia, "Opinion score mining: An algorithmic approach," *International Journal of Intelligent Systems and Applications*, vol. 9, no. 11, pp. 34–41, 2017.

[21] A. Sharaff, C. Kamal, S. Porwal, S. Bhatia, K. Kaur *et al.,* "Spam message detection using danger theory and krill herd optimization," *Computer Networks*, vol. 199, pp. 108421–108453, 2021.

[22] Z. Li, Z. Peng, S. Tang, C. Zhang, H. Ma *et al.,* "Text summarization method based on double attention pointer network," *IEEE Access*, vol. 8, pp. 11279–11288, 2020.

[23] R. Alqaisi, W. Ghanem and A. Qaroush, "Extractive multi-document arabic text summarization using evolutionary multi-objective optimization with k-medoid clustering," *IEEE Access*, vol. 8, pp. 228206–228224, 2020.

[24] S. Ghodratnama, A. Beheshti, M. Zakershahrak and F. Sobhanmanesh, "Extractive document summarization based on dynamic feature space mapping," *IEEE Access*, vol. 8, pp. 139084–139095, 2020.

[25] X. Qian, Y. Wu, M. Li, Y. Ren, S. Jiang *et al.,* "Last: Location-appearance-semantic-temporal clustering-based poi summarization," *IEEE Transactions on Multimedia*, vol. 23, pp. 378–390, 2020.

[26] F. You, S. Zhao and J. Chen, "A topic information fusion and semantic relevance for text summarization," *IEEE Access*, vol. 8, pp. 178946–178953, 2020.

[27] M. Jang and P. Kang, "Learning-free unsupervised extractive summarization model," *IEEE Access*, vol. 9, pp. 14358–14368, 2021.

[28] W. Liu, Y. Gao, J. Li and Y. Yang, "A combined extractive with abstractive model for summarization," *IEEE Access*, vol. 9, pp. 43970–43980, 2021.

[29] S. G. Jindal and A. Kaur, "Automatic keyword and sentence-based text summarization for software bug reports," *IEEE Access*, vol. 8, pp. 65352–65370, 2020.

[30] A. Gidiotis and G. Tsoumakas, "A divide-and-conquer approach to the summarization of long documents," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 3029–3040, 2020.

[31] W. Li and H. Zhuge, "Abstractive multi-document summarization based on semantic link network," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 1, pp. 43–54, 2021.

[32] Q. Zhou, N. Yang, F. Wei, S. Huang, M. Zhou *et al.,* "A joint sentence scoring and selection framework for neural extractive document summarization," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 671–681, 2020.

[33] Y. Gao, Y. Xu, H. Huang, Q. Liu, L. Wei *et al.,* "Jointly learning topics in sentence embedding for document summarization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 4, pp. 688–699, 2019.

[34] P. Zhou, "Character-oriented video summarization with visual and textual cues," *IEEE Transactions on Multimedia*, vol. 22, no. 10, pp. 2684–2697, 2020.

[35] J. Ding, Y. Li, H. Ni and Z.Yang, "Generative text summary based on enhanced semantic attention and gain-benefit gate," *IEEE Access*, vol. 8, pp. 92659–92668, 2020.

[36] M. Yang, C. Li, Y. Shen, Q. Wu, Z. Zhao *et al.,* "Hierarchical human-like deep neural networks for abstractive text summarization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 6, pp. 2744–2757, 2021.

[37] P. Wang, S. Li, H. Zhou, J. Tang, T. Wang *et al.,* "TOC-RWG: Explore the combination of topic model and citation information for automatic related work generation," *IEEE Access*, vol. 8, pp. 13043–13055, 2019.