Research Article

International Journal of Advanced Technology and Engineering Exploration, Vol 12(126)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2024.111100295

Hybrid feature extraction and optimized classification for handwritten Devanagari character recognition using ATOA and BO-SVM

Shubham Srivastava[1*], Ajay Verma[2] and Shekhar Sharma[3]
Research Scholar, Institute of Engineering and Technology, Devi Ahilya Vishwavidyalaya, Indore, India[1]
Professor and Head Electronics and Instrumentation Engineering, Institute of Engineering and Technology, Devi Ahilya Vishwavidyalaya, Indore, India[2]
Professor Electronics and Telecommunication Engineering, Shri Govindram Seksaria Institute of Technology and Science, Indore, India[3]

Received: 14-March-2024; Revised: 04-May-2025; Accepted: 12-May-2025

Abstract
Handwritten character recognition (HCR) is a critical yet challenging task in the fields of computer vision and pattern recognition. It involves the automatic interpretation and classification of characters written by hand, which often vary significantly in style, orientation, and quality. This study proposes a comprehensive methodology aimed at significantly enhancing the accuracy of HCR. The approach comprises several integrated stages. Initially, precise row and column segmentation techniques are employed to isolate individual characters from the input images. A novel hybrid feature extraction method is then introduced, incorporating multiple techniques: speeded-up robust features (SURF) for local feature detection, Gabor filters for capturing texture and edge information, zone transformation for extracting region-specific features, and the radon transform for encoding orientation-related details. To further improve recognition performance, the arithmetic-trigonometric optimization algorithm (ATOA) is utilized for feature selection, enabling the identification of the most relevant features for classification. Subsequently, a support vector machine (SVM) classifier, optimized using Bayesian optimization (BO), is trained to achieve superior classification accuracy through fine-tuned hyperparameters. The proposed methodology demonstrates exceptional performance, with the BO-SVM classifier attaining a classification accuracy of 98.53%. These results highlight the effectiveness of the approach across various character recognition tasks, surpassing the performance of traditional techniques. In addition to its high accuracy, the proposed methodology offers a systematic framework to address the diverse challenges inherent in handwritten character recognition.

Keywords
Handwritten character recognition, Feature extraction, Support vector machine, Bayesian optimization, Arithmetic-trigonometric optimization.

1.Introduction
Character recognition (CR) plays a crucial role in various modern technologies, particularly in artificial intelligence, pattern recognition, and computer vision [1–5]. CR can be classified into two main categories: offline recognition, which deals with character identification from static images, and online recognition, which focuses on real-time recognition during writing [6–8]. Efficient CR systems require extensive datasets and advanced algorithms to achieve accurate results. Among various scripts, Devanagari stands out due to its complexity. It is used in languages like Hindi, Sanskrit, Nepali, and Marathi, featuring 11 vowels, 33 consonants, and additional modifiers, called Matras, that change the characters' forms. Additionally, the Shirorekha, or headline, which connects characters within a word, adds a further layer of intricacy to the recognition process [9–13].

Despite significant progress in handwritten character recognition (HCR), achieving both high accuracy and processing speed remains challenging. Variability in handwriting styles, image noise, and Devanagari's unique features, such as Matras and compound characters, complicate accurate recognition [14–16]. In many cases, current systems prioritize accuracy,

*Author for correspondence

often at the expense of real-time performance. Furthermore, there is a need for improved optimization of preprocessing, feature extraction, and classification techniques for complex scripts like Devanagari.

The main objective of this paper is to enhance the accuracy and efficiency of handwritten Devanagari CR. A novel methodology is proposed to address key challenges in preprocessing, feature extraction, and classification, aiming to balance high recognition accuracy with computational efficiency.

The key contributions of this study are as follows:

- A comprehensive feature extraction framework that integrates speeded-up robust features (SURF), Gabor filters, zone transformation, and the radon transform to capture diverse and discriminative character attributes.
- Implementation of an arithmetic-trigonometric optimization algorithm (ATOA) for effective feature selection, reducing dimensionality while preserving essential characteristics relevant to classification.
- Deployment of a Bayesian-optimized support vector machine (BO-SVM) classifier to enhance recognition accuracy by fine-tuning hyperparameters through Bayesian optimization.

These contributions aim to create a robust and efficient HCR system, particularly for the Devanagari script.

The remainder of the paper is structured as follows: Section 2 reviews relevant literature. The proposed methodology is detailed in section 3, followed by results and discussion in section 4 and section 5 respectively. Section 6 concludes the paper with key findings and future directions.

2.Literature review

The domain of HCR encompasses a broad range of methodologies designed to improve recognition accuracy and robustness. These techniques incorporate feature extraction, classification, and deep learning methods to address the challenges posed by diverse scripts and complex handwriting styles.

Neural network and feature-based methods

Neural networks (NN) were employed in [17] for segmenting and recognizing Devanagari characters by integrating geometric features from skeletonized images with statistical properties such as pixel density. This approach improved character identification performance; however, it was hindered by the complexity of manual feature engineering, which required substantial computational resources and expert intervention, ultimately increasing the overall time and cost of development.

In contrast, [18] utilized a convolutional neural network (CNN) for optical character recognition (OCR) of the Devanagari script. CNNs demonstrated strong capabilities in extracting spatial hierarchies within data, leading to high recognition accuracy. Nevertheless, the model struggled to generalize across varied handwriting styles, revealing limitations in training data diversity and adaptability. Furthermore, the reliance on texture-specific patterns restricted the model's effectiveness when applied to diverse scripts and handwriting variations, thereby reducing its applicability in real-world scenarios.

Gabor filters, as demonstrated in [19], effectively detected textures and edges, achieving 85.73% accuracy for Gurmukhi script recognition. However, this approach also faced challenges in generalizability, as its dependency on specific texture patterns limited performance across different writing styles and scripts. To overcome such limitations, [20] introduced a method that combined Gabor filter banks with classifiers such as k-nearest neighbors (KNN), SVM, and probabilistic neural networks (P-NN). This integration led to a notable 98.15% accuracy in script differentiation. Despite its impressive performance, the method introduced significant computational overhead, which posed scalability concerns when applied to large datasets or real-time applications.

Scale-invariant feature transform (SIFT) and Incremental learning

In [21], SIFT features were employed in conjunction with a SVM classifier to segment characters from historical manuscripts without the need for binarization. While the method proved effective, SIFT exhibited sensitivity to variations in illumination and noise, which compromised its robustness under suboptimal imaging conditions. Further advancements were explored in [22], where incremental learning was combined with SIFT features and the expectation-maximization algorithm for object recognition. This approach demonstrated adaptability to new data by continuously updating the learning model. However, despite its potential, the incremental learning framework showed a tendency to overfit during updates, thereby limiting its

practical application in dynamic environments where data distributions frequently change.

Dimensionality reduction and classification

Dimensionality reduction techniques were evaluated in [23], where principal component analysis (PCA) and linear discriminant analysis (LDA) were applied to high-dimensional datasets to enhance classification performance. PCA demonstrated effectiveness in preserving overall data variance; however, it posed the risk of discarding discriminative features that are crucial for accurate classification, particularly in the context of complex scripts such as Devanagari. In contrast, LDA emphasized class separability but may be less effective when class distributions are non-linear. For Devanagari CR, SVM-based approaches reported in [24] achieved an accuracy of 90.75% for printed text and 92.15% for handwritten characters, as indicated in [25]. Nonetheless, these methods often require intensive preprocessing steps, such as Shiro Rekha removal, which can introduce additional computational overhead and increase system complexity, potentially affecting real-time applicability.

Deep learning and transfer learning approaches

Recent advancements in deep learning have substantially enhanced the accuracy of HCR. CNNs and long short-term memory (LSTM) networks, as explored in [26–28], have shown promising results in multilingual text detection. However, these models often struggle to generalize effectively across complex scripts due to the need for extensive and diverse training data. To address this, transfer learning with pre-trained architectures such as InceptionV3, DenseNet, and visual geometry group (VGG) networks was examined in [29], achieving recognition accuracies of approximately 92%. While these models offer reduced training time and improved feature extraction capabilities, their high computational requirements pose significant challenges for real-time deployment. For instance, DenseNet-121, used in [30], employed dense connectivity patterns to enhance feature reuse and gradient flow, yielding strong performance. Nonetheless, its increased model complexity and computational overhead limit scalability. Similarly, the InceptionV3 architecture, as demonstrated in [31], effectively handled diverse features but encountered difficulties related to training complexity and resource-intensive deployment. Overall, despite their high accuracy, the reliance on large datasets and computationally demanding models reduces the practicality of deep learning-based approaches for time-sensitive or resource-constrained applications.

Feature extraction and hybrid methods

Histogram of oriented gradients (HOG) features, when combined with artificial neural networks (ANNs), achieved 84% accuracy for Devanagari text recognition as reported in [32]. While this approach demonstrated effectiveness, its performance was adversely affected by small or imbalanced datasets, which often led to biased predictions and reduced model reliability. Further improvements were observed in [33], where HOG features were integrated with multi-layer perceptron (MLP) and SVM classifiers, resulting in an improved accuracy of 87%. However, the dependence on manual feature extraction processes limited the model's adaptability to more complex or diverse datasets. In addition, wavelet-based techniques, as presented in [34], achieved a higher accuracy of 93% for Devanagari numeral recognition by leveraging hierarchical feature representations. Despite this success, the interpretability of wavelet-derived features posed a significant challenge, complicating both model transparency and decision-making. Similarly, the use of invariant moments in [35] yielded 82% accuracy for characters and 97% for numerals but exhibited high sensitivity to noise and variations in handwriting styles, which constrained its robustness in practical applications.

Advanced architectures and comparative analyses

The study presented in [36] introduced a deep convolutional neural network (DCNN) with layer-wise learning for CR, achieving an accuracy of 95%. While the model demonstrated strong performance, its heavy dependence on large, labeled datasets for training constrained its generalization capabilities and posed significant challenges for deployment in resource-limited environments. In a related effort, [37] proposed a multilingual CNN-based approach that attained individual language accuracies of 95.54% for English, 93.60% for Hindi, and 91.58% for Bengali, resulting in a combined accuracy of 92.47%. However, the necessity to train separate models for each language undermined the scalability of the approach, particularly in applications involving diverse writing styles or multilingual content. Additionally, the integration of the integral histogram of oriented displacement (IHOD) as a shape descriptor in [38] enhanced classification accuracy to 97.45% by combining local and global text features with a MLP classifier. Despite its high discriminative capability, the method’s reliance on a limited portion of global structural information restricted its effectiveness in handling complex handwriting

patterns and noisy data, thereby limiting its robustness in real-world scenarios.

The reviewed literature illustrates a diverse array of methods and advancements in HCR, reflecting both significant achievements and notable challenges. Techniques such as CNNs and transfer learning have substantially improved recognition accuracy and efficiency across various scripts. However, many methods are constrained by high computational demands, susceptibility to overfitting, and limitations in generalizing across different handwriting styles. NN-based approaches, including CNNs and deep learning models like VGG and DenseNet, have demonstrated impressive results but often at the cost of increased computational requirements. Transfer learning with pre-trained models, such as AlexNet and Inception, has proven effective but still faces challenges related to computational expense and model generalization. Feature extraction techniques, including Gabor filters, HOG, and SIFT, have shown their strengths in specific contexts but also their limitations in handling diverse handwriting styles and large-scale documents. Methods that combine feature extraction with advanced classifiers, such as SVMs and MLPs, have achieved high accuracies but often involve complex preprocessing and substantial computational overhead.

The proposed research paper aims to address these challenges by integrating advanced techniques into a unified methodology. It starts with precise row and column segmentation to improve character isolation, essential for varied writing styles. The hybrid feature extraction method combines SURF for local features, Gabor filters for texture, zone transformation for region-specific attributes, and radon transform for orientation encoding. The ATOA is employed to enhance feature selection, while the BO-SVM classifier is used to refine the recognition process. This integrated approach seeks to balance accuracy with computational efficiency, addressing current limitations and setting a new benchmark in HCR.

3.Proposed methodology

The proposed approach for HCR involves a multi-step process to enhance accuracy. *Figure 1* shows the block diagram for the proposed approach. Initially, the input images undergo row and column segmentation to isolate individual characters. Subsequently, a hybrid feature extraction strategy is employed, combining techniques such as SURF for local feature detection, Gabor filters for texture and edge information, zone transformation for region-specific attributes, and radon transform for orientation details. To streamline the recognition process, feature selection using ATOA optimization identifies the most relevant features. The final step involves training a BO-SVM classifier with optimized hyperparameters for accurate character classification.

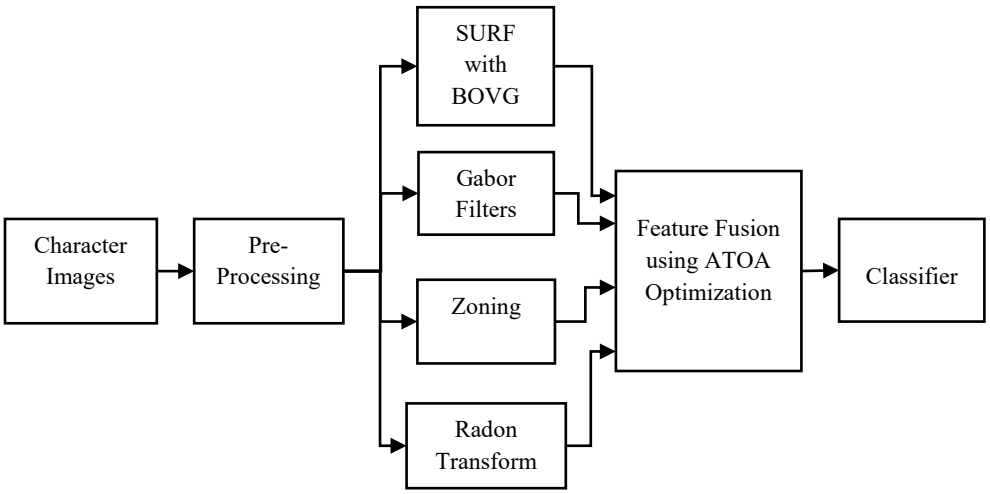

Figure 1 Proposed character recognition with hybrid features

3.1Pre-processing

The application of pre-processing techniques to word or character images serves the purpose of enhancing recognition accuracy by normalizing the images and reducing noise in the signal. Binarization, which converts images to black and white, simplifies the image and highlights crucial features. Normalization, on the other hand, standardizes image appearances,

reducing variations between characters and words, thereby enhancing recognition reliability. While pre-processing is not obligatory, its utilization has demonstrated its potential to enhance recognition outcomes. Additionally, some pre-processing steps aid in feature extraction [5].

Another crucial phase of pre-processing is segmentation, which extracts relevant information for recognition. In the context of this research, segmentation is achieved through the strategic separation of rows and columns within the image [39, 40]. This process isolates individual characters or words, enabling subsequent analysis and classification. Segmentation prepares the image data for recognition tasks, improving accuracy and efficiency.

3.2Rows and column-wise segmentation

The provided information is structured as a symmetrical matrix denoted by $Y = (Y_{ij})_{1\le i,j\le n}$, with a size of $n \times n$. In this arrangement, each cell Y_{ij} signifies the similarity existing between images i and j. It is postulated that there exist K^* divisions in rows (and consequently in columns) labeled as $1 \le t_1^* < \cdots < t_k^* \le n-1$, signifying the boundaries encompassing the aspired $(K^*+1)^2$ segments. For any given pair (k, l) belonging to the set $\{1, \ldots, K^*+1\}^2$, we denote the block R_{kl}^* formulated by [41] in Equation 1:

$$R_{kl}^* = \left\{(i,j) \in \{1,\ldots,n\}^2 \middle| \begin{matrix} t_{k-1}^* + 1 \le i \le t_k^* \\ \text{and } t_{l-1}^* + 1 \le j \le t_l^* \end{matrix}\right\} \quad (1)$$

With the convention that $t_0^* = 0$ and $t_{K^*+1}^* = n$.

3.3Feature extraction

Feature extraction plays a critical role in CR, particularly in the context of intricate scripts like Devanagari characters. To effectively capture the diverse aspects of character structure and variations, a pioneering hybrid feature extraction approach is introduced. This approach amalgamates several distinct methods, each serving a unique purpose. SURF is leveraged for pinpointing local features, Gabor Filters are employed to extract texture and edge information, zone transformation is harnessed to encapsulate region-specific attributes, and the radon transform is utilized to accurately capture orientation details. By combining these techniques, the hybrid approach offers a holistic representation of character features, significantly enhancing the recognition process for complex scripts like Devanagari.

3.3.1Feature description using SURF

SURF is a feature extraction algorithm used to detect and describe keypoints in an image. It's particularly robust to changes in scale, rotation, and illumination.

Keypoint Localization and Refinement

- *Initial Keypoint Detection:* Keypoints are initially detected using the Hessian matrix, which is computed at different scales to identify prominent features. The Hessian matrix is given by Equation 2:

$$H(\sigma) = \begin{bmatrix} D_{xx}(\sigma) & D_{xy}(\sigma) \\ D_{xy}(\sigma) & D_{yy}(\sigma) \end{bmatrix} \quad (2)$$

The determinant of the Hessian matrix helps in identifying keypoints are provided in Equation 3:

$$Det\big(H(\sigma)\big) = D_{xx}(\sigma) \cdot D_{yy}(\sigma) - \big(0.9 \cdot D_{xy}(\sigma)\big)^2 \quad (3)$$

- *Localization Refinement:* To improve the accuracy of keypoint localization, a sub-pixel refinement is applied. This involves fitting a quadratic function to the local region around the keypoint to achieve more precise localization. The refined keypoint location (x', y') is computed as per Equation 4:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} - \frac{1}{2} \cdot H^{-1} \cdot \nabla D \quad (4)$$

Where ∇D represents the gradient of the determinant response and H^{-1} is the inverse of the Hessian matrix.

- *Scale Selection:* To accurately determine the scale of the keypoint, the scale-space extrema are examined. Keypoints are detected at multiple scales by analyzing the Laplacian of gaussian (LoG) responses as per Equation 5:

$$LoG(x, y, \sigma) = \frac{\partial^2 G(\sigma)}{\partial x^2} * I(x, y) \quad (5)$$

Keypoints are selected where the response is a local maximum in both the scale and spatial domain.

Orientation Assignment for Rotation Invariance

- *Orientation Assignment:* To ensure rotation invariance, each keypoint is assigned a dominant orientation based on the local gradient directions. The orientation is computed by Equation 6:

$$Orientation(x, y, \sigma) = \arg\max_{\theta}\Big(\sum_i \sum_j \text{Magnitude}\big(x_i, y_j, \theta\big)\Big) \quad (6)$$

Where $\text{Magnitude}\big(x_i, y_j, \theta\big)$ is the gradient magnitude in direction θ, and the summation is over all the pixels in the keypoint neighborhood.

- *Orientation Histogram:* A histogram of gradient orientations is computed in a region around each keypoint. The peak in this histogram determines the orientation assigned to the keypoint.

Descriptor Construction

- *Subregion Formation:* Around each keypoint, the region is divided into a 4×4 grid, resulting in 16 subregions. Each subregion captures local texture information around the keypoint.
- *Haar Wavelet Responses:* For each subregion, Haar wavelet responses are calculated in both x and y directions. The Haar wavelet responses D_x and D_y are computed as shown in Equations 7 and 8:

$$D_x = I_i\left(x + \frac{w}{2}, y\right) - I_i\left(x - \frac{w}{2}, y\right) \quad (7)$$

$$D_y = I_i\left(x, y + \frac{w}{2}\right) - I_i\left(x, y - \frac{w}{2}\right) \quad (8)$$

Where w is the size of the subregion.

- *Descriptor Generation:* The feature descriptor is formed by concatenating the Haar wavelet responses from all subregions. This results in a high-dimensional vector is presented in Equation 9:

$$Descriptor = [D_{x1}, D_{y1}, D_{x2}, D_{y2}, \ldots, D_{x16}, D_{y16}] \quad (9)$$

Where D_{xi} and D_{yi} are the Haar wavelet responses for each subregion i.

- *Normalization:* The descriptor is normalized to reduce sensitivity to changes in illumination. Equation 10 involves L2 normalization:

$$Normalized\ Descriptor = \frac{\text{Descriptor}}{\|\text{Descriptor}\|_2} \quad (10)$$

Bag of Words Model

- *Descriptor Clustering:* The descriptors obtained from all images are clustered using the K-means algorithm to create a set of visual words. The clustering is performed as follows in below Equation 11:

$$V\{v_1, v_2, \ldots, v_k\} \quad (11)$$

Where k is the number of clusters (visual words) and V represents the set of visual words obtained from the clustering.

- *Histogram Representation:* Each descriptor is assigned to the nearest visual word cluster. This assignment results in a histogram of visual words for each image is shown in Equation 12:

$$Histogram = [h_1, h_2, \ldots, h_k] \quad (12)$$

Where h_i denotes the frequency of the i^{th} visual word in the image.

- *Feature Vector Construction:* The histograms of visual words for each image are used to form the final feature vectors. This approach reduces the dimensionality of the feature space and represents each image by a histogram of visual words. The histograms are normalized to ensure consistency across images with varying sizes and descriptor counts shown in Equation 13:

$$Normalized\ Histogram = \frac{Histogram_i}{\text{sum}(Histogram_i)} \quad (13)$$

It ensures that histograms are scaled between 0 and 1, making them invariant to differences in image size and descriptor density, and thus improving the robustness of the feature representation against illumination changes and other variations.

3.3.2Feature extraction using gabor filters

Gabor filters are designed to capture local frequency and orientation information in an image. The Gabor filter stands as a composite sinusoidal plane oscillation modified by a Gaussian envelope. The universal formula for a two-dimensional (2D) Gabor filter within the spatial realm is represented by the Equation 14:

$$G(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}} e^{i\left(2\pi\frac{x'}{\lambda} + \psi\right)} \quad (14)$$

Where:

- x, y are the spatial coordinates.
- $x' = x\cos(\theta)\, y\sin(\theta), \quad y' = -x\sin(\theta) + y\cos(\theta)$ are rotated coordinates.
- λ is the wavelength of the sinusoidal factor.
- θ is the orientation of the normal to the parallel stripes of the Gabor function.
- ψ is the phase offset.
- σ is the standard deviation of the Gaussian envelope.
- γ is the spatial aspect ratio.

The following are the parameter selection criterion for the Gabor filter:

- Wavelength λ: The wavelength λ of the sinusoidal factor in Gabor filters determines the frequency of the sinusoidal wave. It affects how well the filter captures texture and edge information. Typical values are chosen based on the expected texture size in the character images. For handwritten characters, wavelengths between 2 to 10 pixels are often effective.

- Orientation θ: The orientation θ specifies the direction of the sinusoidal wave relative to the image axes. Multiple orientations (e.g., 0°, 45°, 90°, 135°) are generally used to capture features at different angles. The choice of orientations ensures comprehensive coverage of directional features in the text.
- Phase Offset ψ: The phase offset ψ is often set to 0 or π/2 to control the phase shift of the sinusoidal wave. Common practice is to use multiple phase offsets to capture different textural characteristics.
- Standard Deviation σ (Gaussian Envelope): The standard deviation σ of the Gaussian envelope determines the filter's bandwidth. It is usually set proportional to the wavelength λ. For instance, if λ is set to 4 pixels, a common choice for σ might be $\frac{\lambda}{2}$.
- Spatial Aspect Ratio γ: The aspect ratio γ controls the shape of the Gaussian envelope. A typical value is $\gamma = 1$ for isotropic filters, but values less than 1 or greater than 1 can be used to capture elongated features.
- Basis for Selection: These parameters are often selected based on the nature of the handwritten characters and empirical performance on validation datasets. A parameter optimization approach, such as grid search or cross-validation, is commonly used to find optimal values.

Convolution with Gabor filters

The application of Gabor filters involves the convolution with the image to produce responses characteristic of Gabor filters. The process of convolution in the spatial domain corresponds to the operation of multiplication within the frequency domain as presented in Equation 15.

$$Gabor\ Response\ (x, y; \lambda, \theta, \psi, \sigma, \gamma) = Image(x, y) * G(x, y; \lambda, \theta, \psi, \sigma, \gamma) \quad (15)$$

In the case of each individual image, the Gabor filter reactions are harnessed to produce an assemblage of feature vectors. The constitution of these feature vectors can be achieved by capturing the magnitude of the Gabor filter responses at every pixel location or by calculating statistics within predefined segments. Determine the magnitude of Gabor filter responses as per Equation 16:

$$Magnitude(x, y) = |GaborResponse(x, y)| \quad (16)$$

Divide the image into blocks and calculate statistics, such as mean and variance, for the magnitudes in each block.

3.3.3Zoning

Zoning involves dividing the character image into smaller non-overlapping zones and analyzing the distribution of pixel intensities or other features within each zone. This helps capture local variations in the character's shape and texture. For simplicity, let's assume we divide the character image into a grid of $M \times N$ zones.

Let the character image be represented by the matrix $I(x, y)$, where (x, y) represents the pixel coordinates. Each zone's pixel intensity distribution can be represented by a histogram, which can then be used as a feature vector. The histogram for zone (i, j) can be calculated using the following Equation 17:

$$H_{ij}(k) = \sum_{x=x_{start}}^{x_{end}} \sum_{y=y_{start}}^{y_{end}} \delta(I(x, y) - k) \quad (17)$$

Where:

- $H_{ij}(k)$ is the histogram bin count for intensity value k in zone (i, j).
- $\delta(x)$ is the Dirac delta function, which equals 1 if $x = 0$ and 0 otherwise.
- (x_{start}, y_{start}) and (x_{end}, y_{end}) define the pixel range of the zone (i, j).
- The resulting histograms for all zones will form the feature vector for the Zoning step.

3.3.4Radon transform

The radon transform is a mathematical operation that captures the distribution of intensities along different angles in the image. It is particularly useful for capturing the orientation and slant of characters. The radon transform of an image $I(x, y)$ at angle θ and distance ρ is given by Equation 18:

$$R(\theta, \rho) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(x, y)\delta(x\cos(\theta) + y\sin(\theta) - \rho)dxdy \quad (18)$$

Where:

- $R(\theta, \rho)$ is the radon transform at angle θ and distance ρ.
- $\delta(x)$ is the Dirac delta function.
- θ is the angle of projection.
- ρ is the perpendicular distance from the origin.
- The Radon Transform provides a projection profile for each angle θ, capturing the variation of pixel intensities along that angle.

Selection of derivatives for radon transform:

- Angle θ: The angle θ specifies the projection angles for the radon Transform. A set of angles (e.g., every 1 or 5 degrees from 0° to 180°) is used to capture the orientation of features in different directions.

- Distance ρ: The distance ρ represents the perpendicular distance from the origin in the radon space. It is typically chosen based on the size of the input images. The distance range is generally set to cover the width of the character images, ensuring that projections are fully captured.
- Basis for Selection: The angles are chosen to ensure complete coverage of potential character orientations. The distance ρ is selected based on the dimensions of the image to ensure that the transform covers all possible projections.

3.3.5Concatenation of SURF, Gabor features, zoning feature vectors and radon transform

In our proposed methodology for CR, we integrate Gabor filter responses with SURF features as part of a comprehensive feature extraction process that also includes zoning feature vectors and radon transform. This integration aims to enhance the representation of texture, edges, and other structural attributes of characters. Here's a detailed explanation of how Gabor filter responses are combined with SURF features:

SURF for Local Feature Detection:

- Keypoint Detection: SURF detects and describes keypoints based on the Hessian matrix, which identifies prominent features in the image across various scales and orientations.
- Descriptor Computation: SURF descriptors are computed using Haar wavelet responses in a grid around each keypoint. These descriptors capture local texture patterns and gradients, providing a robust representation of keypoints.

Gabor Filters for Texture and Edge Information:

- Filter Responses: Gabor filters are applied to the entire image to extract texture and edge information. These filters capture patterns and details across different orientations and scales.
- Magnitude Calculation: The magnitude of the Gabor filter responses at each pixel location is calculated to obtain detailed texture and edge features.

Integration of Gabor and SURF Features:

- Feature Extraction from SURF: For each keypoint detected by SURF, we compute the SURF descriptors that represent local texture and edge information around the keypoint.
- Feature Extraction from Gabor Filters: The entire image is processed with Gabor filters to capture texture patterns and edge details at a global level. This results in a set of features that describe texture and edge information across the image.
- Feature Concatenation: To integrate these features, we concatenate the SURF descriptors with the Gabor filter responses. The concatenated feature vector $F_{SURF-Gabor}$ can be represented as per Equation 19:

$$F_{SURF-Gabor} = [D_{SURF1}, D_{SURF2}, \dots, D_{SURFn}, G_1, G_2, \dots, G_m] \quad (19)$$

Where D_{SURFi} represents the SURF descriptors for each keypoint, and G_j represents the Gabor filter magnitudes.

Combining with Other Features:

- Zoning Feature Vectors: In addition to the concatenated SURF and Gabor features, zoning feature vectors are included, which capture local intensity distributions within specific regions of the image.
- Radon transform: The radon transform projections, which capture orientation information, are also concatenated to the feature vector.

Unified Feature Vector Construction:

- The final combined feature vector F is constructed by concatenating SURF features, Gabor filter responses, zoning features, and radon transform projections are provided in Equation 20:

$$F = [D_{SURF1}, D_{SURF2}, \dots, D_{SURFn}, G_1, G_2, \dots, G_m, H_1, H_2, \dots, H_k, R(\theta_1, \rho_1), R(\theta_2, \rho_2), \dots, R(\theta_N, \rho_N)] \quad (20)$$

Where H_i are the zoning histograms and (θ_j, ρ_j) are the radon transform projections.

Benefits of Integration:

- Enhanced Texture and Edge Detection: The combination of SURF's local feature descriptors with the global texture and edge information from Gabor filters results in a more comprehensive feature representation.
- Improved Recognition Performance: Integrating these features with zoning and radon transform further enriches the feature vector, capturing a wide range of character attributes and improving recognition accuracy.

By integrating Gabor filter responses with SURF features, and combining them with zoning and radon transform features, our approach ensures a robust and detailed representation of characters, leading to improved accuracy in HCR.

3.4Feature selection using ATOA optimization

The ATOA [42] is a metaheuristic algorithm that is used to select features for handwritten Devnagari CR. The algorithm works by iteratively updating a population of solutions. Each solution is represented by a vector of binary values, where each value indicates whether or not the corresponding feature is selected. In each iteration, the algorithm does the following:

1. Creates a new solution by perturbing one of the existing solutions.
2. Evaluates the fitness of the new solution using the ATOA objective function.
3. Selects the new solution if it has a better fitness than the existing solution.
4. Repeats steps 2-3 until a stopping criterion is met.

The ATOA objective function is designed to select a set of features that maximizes the recognition accuracy. The objective function is defined in Equation 21 as follows [42]:

$$f(X) = \sum_{i=1}^{N} y_i \log\big(p(y_i|X)\big) \quad (21)$$

Where:
- X is the set of selected features.
- N is the number of classes.
- y_i is the ground truth label for the i^{th} character.
- $p(y_i|X)$ is the probability of the i^{th} character being correctly classified given the set of selected features X.

The ATOA algorithm has been shown to be effective in selecting features for handwritten Devnagari CR. Here are the mathematical equations for the ATOA algorithm.

Initialization: The population of solutions is initialized randomly. Each solution is a vector of binary values, where each value indicates whether or not the corresponding feature is selected as shown in Equation 22.

$$X_i(0) \sim \{0,1\}^n, \; i = 1,2,\dots,N \quad (22)$$

Where:
- N is the population size.
- n is the number of features.

Perturbation: A new solution is created by perturbing one of the existing solutions. The perturbation is done by randomly flipping one of the binary values in the solution as per Equation 23.

$$X_t = X_i \oplus u \quad (23)$$

Where:
X_t is the new solution.
X_i is the existing solution.
u is a random vector of binary values.

Fitness Evaluation: The fitness of the new solution is evaluated using the ATOA objective function. The objective function is calculated using the ground truth labels and the probabilities of the characters being correctly classified in Equation 24.

$$f(X_t) = \sum_{i=1}^{N} y_i \log\big(p(y_i|X_t)\big) \quad (24)$$

Where:
- y_i is the ground truth label for the i^{th} character.
- $p(y_i|X_t)$ is the probability of the i^{th} character being correctly classified given the set of selected features X_t.

Selection: The new solution is selected if it has a better fitness than the existing solution. The solution with the best fitness is selected as the final solution as per Equation 25.

$$f(X_t) > f(X_i) \quad (25)$$

Stopping Criterion: The algorithm terminates if a stopping criterion is met, such as a maximum number of iterations or a minimum fitness value.

Additional Notes:
- The population size, N, is a user-defined hyperparameter.
- The probability distribution for generating the random vector u is a user-defined hyperparameter.
- The stopping criterion is also a user-defined hyperparameter.

3.5Classifier

BO-SVM offers significant advantages in the realm of CR. This approach combines the power of SVM, a robust classification algorithm, with Bayesian optimization, a sophisticated search technique. In the context of CR, where accuracy is paramount, BO-SVM offers a tailored solution for identifying optimal hyperparameters. Bayesian optimization efficiently explores the SVM's hyperparameter space to identify configurations that improve CR performance. This involves iteratively selecting hyperparameters based on probabilistic predictions, evaluating accuracy, and refining the model. This dynamic process ensures that the SVM is fine-tuned to extract the most relevant features and make accurate character classifications. Complex scripts like Devanagari require a comprehensive approach, and BO-SVM enhances recognition accuracy by balancing parameters such as kernel type and regularization. As a result, BO-SVM could revolutionize CR tasks, especially for intricate scripts, contributing to more accurate and efficient recognition systems.

Classifier training and testing using concatenated features

Classifier Training: The concatenated feature vectors are used to train a classifier for CR as presented in Equation 26.

$$Classifier = Train\ Classifier\ (Combined\ Feature\ Vector, Labels \quad (26)$$

Classifier Testing: The trained classifier is used to recognize characters in new images based on their concatenated feature vectors, shown in Equation 27.

$$Predicted\ Labels = Classifier(Patterns) \quad (27)$$

By combining the strengths of selected features approach aims to capture a comprehensive range of information from the images, including both local structure and textural features. This approach can lead to improved CR accuracy.

The steps of the approach are shown below.

Preprocessing:
- *Input: Image*
- *Output: Segmented characters*
- *Steps:*
  a. *Convert to grayscale*
  b. *Binarize and denoise*
  c. *Perform row and column segmentation*

Feature Extraction:
- *Input: Segmented characters*
- *Output: Feature vector*
- *Steps:*
  a. *Extract local features using SURF*
  b. *Extract texture using Gabor filters*
  c. *Extract region-based features using Zoning*
  d. *Extract orientation using radon transform*
  e. *Concatenate all features into a single vector*

Feature Selection (ATOA):
- *Input: Feature vectors*
- *Output: Optimized features*
- *Steps:*
  a. *Initialize population of binary solutions (selected features)*
  b. *Evaluate and update solutions based on fitness (SVM accuracy)*
  c. *Repeat until stopping criteria are met*

Classification (BO-SVM):
- *Input: Optimized features*
- *Output: Predicted labels*
- *Steps:*
  a. *Train SVM with Bayesian Optimization to tune hyperparameters*
  b. *Test on dataset and evaluate performance (accuracy, precision)*

Final Output:
- *Input: Predicted labels*
- *Output: Recognized characters*
- *Steps:*
  a. *Map predicted labels to characters*
  b. *Output recognized characters*

4.Results

4.1Evaluation parameters

Table 1 shows the evaluation parameters used in this research, based on which the accuracy, precision, sensitivity, and F1-score are calculated.

Table 1 Evaluation parameters

TP (True Positive)	"Indicated the number of characters that were classified as correctly classified"
TN (True Negative)	"Indicated the number of characters that were classified as not classified correctly"
FP (False Positive)	"Indicated the number of characters that were classified as incorrectly classified"
FN (False Negative)	"Indicated the number of characters that were classified as not Classified incorrectly"

These parameters (*Table 1*) are essential for evaluating the performance of classification models and are used to derive key metrics as follows (Equations 28-31):

Accuracy = (TP + TN) / (TP + TN + FP + FN) (28)

Precision = TP / (TP + FP) (29)

Sensitivity (Recall) = TP / (TP + FN) (30)

F1-Score = 2 × (Precision × Recall) / (Precision + Recall) (31)

4.2Dataset

4.2.1Handwritten Marathi Devanagari characters dataset

The Handwritten Marathi Devanagari Characters dataset, available on Kaggle, comprises 58 distinct handwritten Devanagari characters, including 12 vowels, 36 consonants, and 10 numerals. The dataset was curated by Shalaka Deore and collected from individuals across various age groups to ensure diversity in handwriting styles. Each character is represented by 100 samples, resulting in a total of 5,800 grayscale images. All images are uniformly sized at a resolution of 28×28 pixels [43].

4.2.2Devanagari handwritten character dataset (DHCD) UCI dataset

The DHCD showcasing handwritten Devanagari characters, accessible through University of California, Irvine (UCI), is comprised of 92000 monochromatic images of Devanagari characters that were scribed by 46 distinct individuals. This collection encompasses the entire gamut of 36 consonants, 14 vowels, and 10 numerals featured within the Devanagari script. Each image bears dimensions of 32×32 pixels.

4.2.3CPAR database

Character pattern recognition (CPAR) stands as a widely employed repository for research pertaining to the recognition of Devanagari characters. Within its confines reside 35,000 isolated handwritten numerals and an

additional 83,300 characters. This trove has been accumulated from individuals hailing from varied age groups, religious backgrounds, and educational levels. This eclectic assemblage renders the database an accurate reflection of the multifarious handwriting styles encountered in the real world.

4.2.4Dongre and Mankar database

The Dongre and Mankar database boasts a compilation of tagged image file format (TIFF) format images comprising 5137 numerals and 20305 characters, all captured through the contributions of 750 different writers.

4.3Results

Figure 2 shows the comparative analysis of accuracy between SVM and BO-SVM when trained and tested on various datasets reveals a clear advantage for BO-SVM. Across different datasets, the accuracy achieved by BO-SVM consistently surpasses that of traditional SVM. This outcome establishes BO-SVM as the obvious choice due to its consistently higher accuracy rates. This performance improvement highlights the effectiveness of Bayesian optimization in refining the SVM's hyperparameters for better classification outcomes, making BO-SVM a compelling and preferred approach for CR tasks.

Table 2 provides a thorough comparative analysis of the classification performance of BO-SVM across multiple datasets used for CR. The metrics evaluated include accuracy, error rate, sensitivity, specificity, precision, false positive rate, F1-score, Matthews Correlation (MCC), and Kappa statistic. On the Marathi dataset, BO-SVM achieved a high accuracy of 95.83%, with sensitivity and specificity values of 95.83% and 98.61%, respectively. In the DHCD UCI dataset, BO-SVM attained an even higher accuracy of 98.53% and a specificity of 99.51%. For the CPAR database and the Dongre and Mankar database, the BO-SVM demonstrated accuracies of 96.67% and 97.22%, respectively, while maintaining strong performance across precision, F1-score, MCC, and Kappa statistics. These results highlight the effectiveness of BO-SVM in delivering robust and reliable CR performance across a variety of datasets, demonstrating its consistent ability to achieve high accuracy and reliable performance metrics.

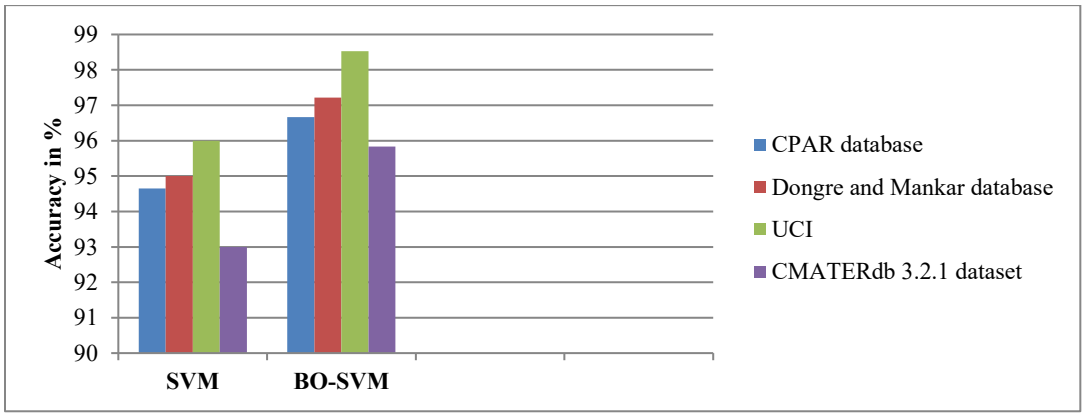

Figure 2 Comparative analysis of accuracy with ATOA optimized hybrid features on various dataset trained and tested with SVM and BO-SVM

Table 2 Comparative analysis of classification result with various dataset on BO-SVM with ATOA optimized hybrid features

Parameters	Handwritten marathi devanagari characters dataset	DHCD UCI Dataset	CPAR Database	Dongre And Mankar Database
Accuracy	0.9583	0.9853	0.9667	0.9722
Error	0.0417	0.0147	0.0333	0.0278
Sensitivity	0.9583	0.9853	0.9667	0.9722
Specificity	0.9861	0.9951	0.9889	0.9907
Precision	0.9623	0.9861	0.9682	0.975
False Positive Rate	0.0139	0.0049	0.0111	0.0093
F1-Score	0.9586	0.9853	0.9669	0.9721
Matthews Correlation Coefficient	0.9462	0.9808	0.9562	0.9642
Kappa	0.8889	0.9608	0.9111	0.9259

Table 3 provides a detailed comparative analysis of classification performance using the BO-SVM with various feature sets for CR. The features evaluated include SURF-BOW with Gabor Filters, GABOR features, zoning-based features, radon transform-based features, and hybrid features. This analysis covers multiple performance metrics such as accuracy, error rate, sensitivity, specificity, precision, false positive rate, F1-score, MCC, and Kappa statistic.

Among the features evaluated, hybrid features achieve the highest accuracy of 98.53%, clearly surpassing other feature sets. SURF features also perform exceptionally well, with an accuracy of 97.5%. The consistently high specificity across all feature sets demonstrates their strong ability to correctly identify negative instances, while the F1-scores for hybrid features and SURF indicate a well-balanced precision-recall performance. The Matthews Correlation Coefficient and Kappa statistics further highlight the robustness of the hybrid features and the superior performance of SURF. These results collectively demonstrate the effectiveness of hybrid and SURF features in delivering high accuracy and robust classification outcomes through the BO-SVM approach, reinforcing their potential for CR tasks.

Table 3 Comparative analysis of classification result with various features on BO-SVM

Parameters	Surf Bow	Gabor	Zoning	Radon transfporm	Hybrid features
Accuracy	9.75E-01	0.9643	0.9417	0.9559	0.9853
Error	2.50E-02	0.0357	0.0583	0.0441	0.0147
Sensitivity	1.00E+00	0.9643	0.9417	0.9559	0.9853
Specificity	9.50E-01	0.9881	0.9806	0.9853	0.9951
Precision	9.52E-01	0.9643	0.9464	0.9625	0.9861
False Positive Rate	5.00E-02	0.0119	0.0194	0.0147	0.0049
F1-Score	9.76E-01	0.9643	0.9421	0.957	0.9853
Matthews Correlation Coefficient	9.51E-01	0.9524	0.9243	0.9442	0.9808
Kappa	9.50E-01	0.9048	0.8444	0.8824	0.9608

Table 4 provides a comparative analysis of various advanced methods for HCR, highlighting the performance of state-of-the-art approaches. Previous research has employed a range of techniques and datasets. For instance, Narang et al. [16] utilized a combination of MLP, NN, CNN, decision trees, and random forests on Devanagari ancient text, achieving an accuracy of 88%. Gaikwad et al. [44] achieved 93.6% accuracy using discrete cosine transform (DCT) and geometric features with LDA on Devanagari characters. Sahare et al. [45] excelled with the CPAR dataset by employing feature component distribution function (FCDDF), feature component curvature function (FCCF), and normalized curvature features (NCF) combined with a statistical neural KNN classifier, attaining an impressive accuracy of 98.26%. Reddy and Babu [23] reached a 95.57% accuracy on the UCI dataset using CNN for feature extraction and a deep feed-forward NN for classification. In contrast, our proposed method integrates a novel hybrid feature extraction approach that combines SURF, Gabor filters, zoning, and radon transform, along with an ATOA-optimized feature selection algorithm. When applied to the UCI, Dongre and Mankar, CPAR, and Marathi datasets, this method demonstrates superior performance with accuracies of 98.53%, 97.22%, 96.67%, and 95.83%, respectively. This consistent outperformance across multiple benchmarks underscores the effectiveness and robustness of the proposed BO-SVM approach in enhancing CR accuracy.

Table 4 Comparative analysis of classification result with various state of arts

Authors	Database	Features Extracted	Classifier	Accuracy (%)
Narang et al., [16]	Devanagari ancient text	Intersection and open endpoint, Centroid, Horizontal peak extent, Vertical peak extent	Simple majority voting on MLP, NN,CNN, decision tree, random forest	88%
Gaikwad et al., [44]	Devnagari	DCT, Geometric features	LDA	93.6%
Sahare et al., [45]	CPAR	FCDDF+FCCF+NCF	Statistical Neural KNN	98.26%
Reddy and Babu [23]	UCI	Features extracted by CNN	Deep feed-forward neural network with 2 hidden layers	95.57 %

Authors	Database	Features Extracted	Classifier	Accuracy (%)
Proposed method	UCI	SURF BOVG + Gabor + Zoning, +Radon +ATOA optimized feature selection algorithm .	BO-SVM	98.53%
Proposed method	Dongre and Mankar dataset	SURF BOVG + Gabor + Zoning, + Radon + ATOA optimized feature selection algorithm .	BO-SVM	97.22%
Proposed method	CPAR dataset	SURF BOVG + Gabor + Zoning, + Radon + ATOA optimized feature selection algorithm .	BO-SVM	96.67%
Proposed method	Handwritten Marathi Devanagari Characters Dataset	SURF BOVG + Gabor + Zoning, +Radon + ATOA optimized feature selection algorithm .	BO-SVM	95.83%

5.Discussion

The results presented in this study highlight the significant advantages of the BO-SVM approach for CR tasks. As illustrated in *Figure 2* and detailed in *Tables 2* and *3*, BO-SVM consistently outperforms conventional SVM models across various datasets and feature sets. These findings confirm the effectiveness of Bayesian optimization in fine-tuning SVM hyperparameters, resulting in enhanced classification accuracy and overall model performance. Specifically, the BO-SVM achieved accuracies ranging from 95.83% to 98.53%, surpassing those of traditional SVM techniques. Beyond improved accuracy, the model also demonstrated strong performance across sensitivity, specificity, precision, and F1-score metrics. The integration of a hybrid feature extraction strategy—combining SURF, Gabor filters, zoning, and Radon transform—contributed notably to these outcomes, with the highest recorded accuracy reaching 98.53%.

The robustness of the BO-SVM model and the effectiveness of hybrid feature extraction techniques highlight its strong potential for CR tasks. The ability to integrate advanced feature extraction methods with Bayesian optimization has proven to substantially enhance classification outcomes. This makes the BO-SVM approach particularly valuable in practical applications that demand high accuracy, such as automated document processing and digital text recognition systems, where precise character classification is critical.

A comparative analysis with state-of-the-art methods reveals that the proposed approach surpasses existing CR technologies. While previous methods, such as those by Narang et al. [16], Gaikwad et al. [44], Sahare et al. [45], and Reddy et al. [23], achieved notable results, they were outperformed by the BO-SVM model. The combination of hybrid features and Bayesian optimization consistently provided superior accuracy across various datasets. This comparative advantage demonstrates the potential of the BO-SVM approach to significantly advance the field of CR, suggesting that this methodology can provide a robust solution for handling complex CR tasks in diverse applications.

5.1Limitations

While the BO-SVM approach demonstrates strong performance, several limitations are evident. The datasets used in this study may not fully capture the range of handwriting variations encountered in real-world settings, including differences influenced by demographic and cultural factors. This could affect the model's ability to generalize effectively. Additionally, the computational complexity of the BO-SVM method has not been addressed, which may pose challenges for implementation in resource-constrained environments. The model's generalization capability to unseen data has not been thoroughly assessed, limiting insights into its practical deployment in diverse scenarios. Moreover, the performance of the model under noisy conditions or when characters are occluded has not been extensively examined, which may impact its robustness in real-world applications.

A complete list of abbreviations is listed in *Appendix I.*

6.Conclusion and future work

This paper presents an innovative methodology that advances the field of HCR by achieving substantial improvements in accuracy. The strength of the proposed approach lies in its systematically coordinated sequence of stages, each contributing to enhanced recognition performance. The process begins with effective character segmentation using row and column isolation, providing a reliable structural foundation. This is followed by the introduction of a hybrid feature extraction technique that combines SURF, Gabor filters, zone transformation, and radon transform to capture comprehensive character descriptors. Feature selection is further refined through the ATOA

optimization algorithm, which emphasizes the most discriminative features to improve classification outcomes. The proposed framework demonstrates superior performance over existing benchmarks, attaining a maximum recognition accuracy of 98.53%. The methodology concludes with the training of a BO-SVM classifier, which utilizes optimized hyperparameters to achieve high-precision character classification. This integrated pipeline ensures both robustness and operational efficiency in HCR applications.

Future research will explore deep learning-based feature extraction and classification techniques using CNNs to further enhance recognition accuracy. Additionally, transfer learning approaches will be examined to improve generalization across diverse and heterogeneous datasets. Optimization of the system for real-time handwritten CR in resource-constrained environments will also be a key focus to ensure both efficiency and practical applicability.

Acknowledgment
The authors would like to express their gratitude to Institute of Engineering and Technology, Devi Ahilya Vishwavidyalaya, Indore, Madhya Pradesh for all of their assistance and encouragement in carrying out this research and publishing this paper.

Conflicts of interest
The authors have no conflicts of interest to declare.

Data availability
The datasets used in this study, which include handwritten Devanagari character images, were collected from publicly available sources such as Kaggle, UCI, CPAR, and Dongre and Mankar databases. These datasets are not proprietary and are freely accessible for research purposes. However, specific access details for each dataset are provided in the respective references.

Author's contribution statement
Shubham Srivastava: Methodology, data collection, data curation, analysis and interpretation of results. **Ajay Verma and Shekhar Sharma:** Conceptualization, investigation, methodology, supervision, validation, investigation on challenges and draft manuscript preparation.

References
[1] Jaiswal S, Chaudhari K, Tade S, Khirwadkar S, Pande A. Empirical review on handwritten Devanagari script recognition techniques using AI approaches. International Journal of Creative Computing. 2024; 2(2):119-37.

[2] Inunganbi S. A systematic review on handwritten document analysis and recognition. Multimedia Tools and Applications. 2024; 83(2):5387-413.

[3] Sharma C, Sharma S, Sakshi, Chen HY. Advancements in handwritten Devanagari character recognition: a study on transfer learning and VGG16 algorithm. Discover Applied Sciences. 2024; 6(12):1-20.

[4] Devendiran S, Nedungadi P, Raman R. Handwritten text recognition using VGG19 and HOG feature descriptors. In international conference on interdisciplinary approaches in technology and management for social innovation 2024 (pp. 1-4). IEEE.

[5] El MM. An effective combination of convolutional neural network and support vector machine classifier for Arabic handwritten recognition. Automatic Control and Computer Sciences. 2023; 57(3):267-75.

[6] Singh S, Garg NK, Kumar M. VGG16: offline handwritten Devanagari word recognition using transfer learning. Multimedia Tools and Applications. 2024; 83(29):72561-94.

[7] Duarte JM, Berton L. A review of semi-supervised learning for text classification. Artificial Intelligence Review. 2023; 56(9):9401-69.

[8] Singh S, Sharma A, Chauhan VK. Indic script family and its offline handwriting recognition for characters/digits and words: a comprehensive survey. Artificial Intelligence Review. 2023; 56(Suppl 3):3003-55.

[9] Jabde KM, Patil HC, Vibhute DA, Mali S. A comprehensive literature review on air-written online handwritten recognition. International Journal of Computing and Digital Systems. 2024; 15(1):307-22.

[10] Sharma A, Mithun BN. Deep learning character recognition of handwritten Devanagari script: a complete survey. In international conference on contemporary computing and communications 2023 (pp. 1-6). IEEE.

[11] Nagane AS, Patil CH, Mali SM. Classification of brahmi script characters using HOG features and multiclass error-correcting output codes (ECOC) model containing SVM binary learners. In international conference on intelligent and innovative technologies in computing, electrical and electronics 2023 (pp. 448-51). IEEE.

[12] Kaur R. An overview of advanced technologies applied to identified printed and handwritten text in Gurmukhi script: a review. Journal Punjab Academy of Sciences. 2023; 23:194-207.

[13] Bose SR, Singh R, Joshi Y, Marar A, Regin R, Rajest SS. Light weight structure texture feature analysis for character recognition using progressive stochastic learning algorithm. In advanced applications of generative AI and natural language processing models 2024 (pp. 144-58). IGI Global Scientific Publishing.

[14] Rajpal D, Garg AR. Deep learning model for recognition of handwritten Devanagari numerals with low computational complexity and space requirements. IEEE Access. 2023; 11:49530-9.

[15] Moudgil A, Singh S, Sareen B, Wadhwa S. Devanagari ancient manuscript recognition using AlexNet. In 11th international conference on reliability, infocom technologies and optimization (Trends and Future Directions) 2024 (pp. 1-5). IEEE.
[16] Narang S, Jindal MK, Kumar M. Devanagari ancient documents recognition using statistical feature extraction techniques. Sādhanā. 2019; 44(6):141.
[17] Khanderao MS, Ruikar S. Character segmentation and recognition of Indian Devanagari script. In ICT analysis and applications: proceedings of ICT4SD 2019 (pp. 529-37). Springer Singapore.
[18] Dessai B, Patil A. A deep learning approach for optical character recognition of handwritten Devanagari script. In 2nd international conference on intelligent computing, instrumentation and control technologies 2019 (pp. 1160-5). IEEE.
[19] Singh S, Aggarwal A, Dhir R. Use of gabor filters for recognition of handwritten Gurmukhi character. International Journal of Advanced Research in Computer Science and Software Engineering. 2012; 2(5):234-40.
[20] Rani R, Dhir R, Lehal GS. Gabor features based script identification of lines within a bilingual/trilingual document. International Journal of Advanced Science and Technology. 2014; 66:1-12.
[21] Diem M, Sablatnig R. Recognizing characters of ancient manuscripts. In computer vision and image analysis of Art 2010 (pp. 35-46). SPIE.
[22] Helmer S, Lowe DG. Object class recognition with many local features. In conference on computer vision and pattern recognition workshop 2004 (pp. 187-7). IEEE.
[23] Reddy RV, Babu UR. Handwritten Hindi character recognition using deep learning techniques. International Journal of Computer Sciences and Engineering. 2019; 7(2):1-17.
[24] Gadekallu TR, Khare N, Bhattacharya S, Singh S, Maddikunta PK, Ra IH, et al. Early detection of diabetic retinopathy using PCA-firefly based deep learning model. Electronics. 2020; 9(2):1-16.
[25] Puri S, Singh SP. An efficient Devanagari character classification in printed and handwritten documents using SVM. Procedia Computer Science. 2019; 152:111-21.
[26] Bhunia AK, Konwer A, Bhunia AK, Bhowmick A, Roy PP, Pal U. Script identification in natural scene image and video frames using an attention based convolutional-LSTM network. Pattern Recognition. 2019; 85:172-84.
[27] Tian S, Bhattacharya U, Lu S, Su B, Wang Q, Wei X, et al. Multilingual scene character recognition with co-occurrence of histogram of oriented gradients. Pattern Recognition. 2016; 51:125-34.
[28] Shi B, Yao C, Zhang C, Guo X, Huang F, Bai X. Automatic script identification in the wild. In 13th international conference on document analysis and recognition 2015 (pp. 531-5). IEEE.
[29] Aneja N, Aneja S. Transfer learning using CNN for handwritten Devanagari character recognition. In 1st international conference on advances in information technology 2019 (pp. 293-6). IEEE.
[30] Huang G, Liu Z, Van DML, Weinberger KQ. Densely connected convolutional networks. In proceedings of the conference on computer vision and pattern recognition 2017 (pp. 4700-8). IEEE.
[31] Talebi H, Milanfar P. Learning to resize images for computer vision tasks. In proceedings of the international conference on computer vision 2021 (pp. 497-506). IEEE/CVF.
[32] Singh N. An efficient approach for handwritten Devanagari character recognition based on artificial neural network. In 5th international conference on signal processing and integrated networks 2018 (pp. 894-7). IEEE.
[33] Jadhav S, Inamdar V. Convolutional neural network and histogram of oriented gradient based invariant handwritten Modi character recognition. Pattern Recognition and Image Analysis. 2022; 32(2):402-18.
[34] Arora S, Bhatcharjee D, Nasipuri M, Malik L. A two stage classification approach for handwritten Devnagari characters. In international conference on computational intelligence and multimedia applications 2007 (pp. 399-403). IEEE.
[35] Ghimire A, Chapagain A, Bhattarai U, Jaiswal A. Nepali handwriting recognition using convolution neural network. International Research Journal of Innovations in Engineering and Technology. 2020; 4(5):1-5.
[36] Gharde SS, Ramteke RJ, Kotkar VA, Bage DD. Handwritten Devanagari numeral and vowel recognition using invariant moments. In international conference on global trends in signal processing, information computing and communication 2016 (pp. 255-60). IEEE.
[37] Kundaikar T, Fadte S, Karmali R, Pawar JD. Automatic Hindi OCR error correction using MLM-BERT. Ingénierie des Systèmes d'Information. 2024; 29(2): 619-26.
[38] Omayio EO, Indu S, Panda J. Word spotting and character recognition of handwritten Hindi scripts by integral histogram of oriented displacement (IHOD) descriptor. Multimedia Tools and Applications. 2024; 83(1):1379-406.
[39] Bisht M, Gupta R. Offline handwritten Devanagari modified character recognition using convolutional neural network. Sādhanā. 2021; 46:1-4.
[40] Wan CH, Lee LH, Rajkumar R, Isa D. A hybrid text classification approach with low dependency on parameter by integrating K-nearest neighbor and support vector machine. Expert Systems with Applications. 2012; 39(15):11880-8.
[41] Mehta N, Doshi J. Segmentation methods: a review. International Journal for Research in Applied Science and Engineering Technology. 2020; 8:536-40.
[42] Panneer SAM, Hussin FA, Ibrahim R, Bingi K. Arithmetic-trigonometric optimization algorithm. In optimal fractional-order predictive PI controllers: for

process control applications with additional filtering 2022 (pp. 99-133). Singapore: Springer Nature Singapore.
[43] Deore SP, Pravin A. Devanagari handwritten character recognition using fine-tuned deep convolutional neural network on trivial dataset. Sādhanā. 2020; 45(1):243.
[44] Gaikwad SS, Nalbalwar SL, Nandgaonkar AB. Devanagari handwritten characters recognition using DCT, geometric and hue moments feature extraction techniques. Sādhanā. 2022; 47(3):112.
[45] Sahare P, Dhok SB. Multilingual character segmentation and recognition schemes for Indian document images**.** IEEE Access. 2018; 6:10603-17.

Shubham Srivastava is currently pursuing his Ph.D in Electronics and Telecommunication from IET DAVV Indore. He has completed his B.E. in Electronics and Communication from IIST Indore affiliated to RGPV, Bhopal and M.E. in Electronics Engineering from IET DAVV Indore. He is currently working as Assistant Professor in Electronics and Telecommunication Engineering department in SGSITS Indore.
Email: shubhamsci2023@gmail.com

Ajay Verma is currently working as Professor and Head Electronics and Instrumentation Engineering department in IET DAVV Indore. He received his Ph.D. degree in year 2002 from Devi Ahilya University, Indore. His research interests include Image Processing, Nonlinear Dynamics and System Theory and Nonlinear Control System.
Email: averma@ietdavv.edu.in

Shekhar Sharma received AMIE from Electronics and Communication from the Institution of Engineers (India), M.E. in Electronics Engineering from Anna University Chennai and Ph.D from School of Electronics, DAVV Indore. At present he is working as Professor in the department of Electronics and Telecommunication engineering at SGSITS Indore. His teaching and research interests include Signal Processing and Digital Communication and Application of Soft Computing in these areas.
Email: gs0400228@sgsitsindore.in

Appendix I

S. No.	Abbreviation	Description
1	ANN	Artificial Neural Network
2	ATOA	Arithmetic-Trigonometric Optimization Algorithm
3	BO	Bayesian Optimized
4	BO-SVM	Bayesian-optimized Support Vector Machine
5	CNN	Convolutional Neural Network
6	CPAR	Character Pattern Recognition
7	CR	Character Recognition
8	DCNN	Deep Convolutional Neural Network
9	DCT	Discrete Cosine Transform
10	DHCD	Devanagari Handwritten Character Dataset
11	FCCF	Feature Component Curvature Function
12	FCDDF	Feature Component Distribution Function
13	HCR	Handwritten Character Recognition
14	HOG	Histogram of Oriented Gradients
15	IHOD	Integral Histogram of Oriented Displacement
16	KNN	K-Nearest Neighbors
17	LDA	Linear Discriminant Analysis
18	LoG	Laplacian of Gaussian
19	LSTM	Long Short-Term Memory
20	MLP	Multi-Layer Perceptron
21	NCF	Normalized Curvature Features
22	NN	Neural Networks
23	OCR	Optical Character Recognition
24	PCA	Principal Component Analysis
25	PNN	Probabilistic Neural Networks
26	SIFT	Scale-Invariant Feature Transform
27	SURF	Speeded-Up Robust Features
28	SVM	Support Vector Machine
29	TIFF	Tagged Image File Format
30	VGG	Visual Geometry Group