Comparative Study of Feature Extraction Techniques for Handwritten Ancient Modi Script Recognition

Ketki R. Ingole[1], Pritish Tijare[2]

[1]Research Scholars, Department of Computer Science and Engineering Sipna College of Engineering and Technology Amravati, India, Email: mohodketki@gmail.com
[2]Professor, Department of Computer Science and Engineering Sipna College of Engineering and Technology Amravati, India, Email: pritishtijare1980@gmail.com

Received: 10.04.2024 Revised : 12.05.2024 Accepted: 22.05.2024

ABSTRACT
In Maharashtra, documentation has been carried out using Devanagari script since the 19th century. Before the adoption of Devanagari, the Modi script was predominantly used for documentation, but its cursive writing style, complexity, and limited awareness led to its decline. The Modi script has deep historical origins, and numerous manuscripts are stored in Museums, National Libraries, and cultural heritage sites, such as temples,forts. Unfortunately, much of the knowledge contained in these documents remains inaccessible due to the lack of proper resources. The Government is working on preserving these documents and revealing the information they contain. Utilization of modern technologies, such as HOCR (Handwritten Optical Character Recognition), is now underway to restore cultural heritage. This study specifically focuses on recognizing script data from image documents, particularly handwritten Modi script. Feature extraction techniques play a crucial role in identifying text from the ancient Modi script, as illustrated by the comparative analysis provided here.

Keywords: cursive, Modi-script, HOCR, Manuscripts, feature extraction

1. INTRODUCTION

Field researchers focus on Optical Character Recognition (OCR), which is a part of Computer Vision and Pattern Recognition. Although there has been significant progress in identifying printed text, recognizing handwritten text remains one of the most difficult challenges due to the variation in individual handwriting styles. The complexity is even greater when it comes to recognizing characters from ancient manuscripts, as these scripts often introduce additional complications. Advancing research in the recognition of ancient scripts is essential for uncovering the knowledge preserved in historical manuscripts.

Extensive study of foreign languages has shown promising breakthroughs in Optical Character Recognition (OCR). In India, however, text recognition offers issues due to regional variances in languages, where spoken and written forms might differ greatly. The complexity of Indian scripts adds to the difficulty of this undertaking. Many scripts have been impacted by historical kingdoms, social contexts, and regional languages. According to a survey, an article [5] states that cursive writing is a significant barrier to Intelligent Character Recognition (ICR), as it is difficult to segment and distinguish letters written in continuous patterns.

In Maharashtra, the regional script Modi, which evolved throughout time under the influence of several governing dynasties, was finally supplanted by the Devanagari script in the late twentieth century. Modi script, the Marathi writing system, has a 700-year history. It began in the 12th century under the Yadav Dynasty, evolved throughout the Bahamani dynasty, and was further perfected during the Chhatrapati Shivaji and Peshwa eras. After India gained independence, the Devanagari script progressively supplanted Modi. The term Modi was also used in Tamil and Gujarati. The purpose of this study is to identify and characterize the fundamental structural elements of Modi script letters, with an emphasis on the components and qualities that distinguish one letter from another and are required for efficient feature extraction.

The Modi script is missing the necessary punctuation and markers to show the start and finish of sentences and words, making it difficult to read. It contains 46 characters: 36 consonants and 10 vowels, in contrast to the Devanagari script, which has 48 characters: 36 consonants and 12 vowels. The reduced number of vowels in the Modi script contributes to fewer mistakes in grammar compared to Devanagari.

The script is created by drawing a line from left to right across the page, known as 'Shirolekh', and placing the letters based on this line.
In the past, the Modi script was extensively used for official and administrative tasks, as well as forreligious writings, books, letters, and more. It was particularly popular during the Maratha Empire andcontinued to be used until the early 1900s.The advent of the British Raj, coupled with the proliferation of printing technology, contributed to a decline in the usage of Modi script, as it was increasinglyfavored over Devanagari and other scripts.

2. LITERATURE REVIEW

Researchers have delved into the recognition of various Indic and medieval scripts, including the Modi script. This script, with its rich historical and literary significance in central and western India, has undergone changes over time, leading to notable variations. Despite its importance, automating Optical Character Recognition (OCR) for the Modi script has not been extensively explored. Initial research by D. N. Besekar et al. [2], [3] has primarily focused on this area. Paper [2] investigates the structural similarities between standard and handwritten Modi characters, while paper [3] offers a theoretical analysis of the script and the challenges it presents in recognition. Their specific efforts encompass mathematical morphological approaches [4] and chain code-based recognition [6]. However, the accuracy of chain codes may vary due to the script's free-form writing style and the similarity of many characters.
Using Otsu's Binarization and the Kohonen neural network technique, Sidra Anam and associates [7] created the Modi Script Character Recognizer System (MSCR). This system showed promise, achieving 72.6 percent accuracy for handwritten characters, but had lower recognition rates for similarly shaped characters. It was trained on 22 different Modi characters from handwritten samples. Using Measured Structure Similarity (SSIM), KNN, and Backpropagation Neural Networks, Ramteke A.S. and Katkar G.S. [8] were able to identify Modi characteristics with a recognition rate of 91 to 97 percent.
Size normalization and noise removal were two preprocessing steps that Besekar D.N. and Ramteke R.J. [9] used to achieve a 93.5 percent accuracy with a variance table when using a zone-based technique for offline handwritten digit recognition. Theoretical analysis of Modi script recognition by Besekar D. N. & Ramteke R. J. [10] highlighted the challenges in extracting structural elements due to the script's cursive form, character variations, and handwriting styles. The study recommended improvements for segmentation and recognition.
Theoretical analysis of Modi script recognition by Besekar D. N. & Ramteke R. J. [10] highlighted the challenges in extracting structural elements due to the script's cursive form, character variations, and handwriting styles. The study recommended improvements for segmentation and recognition.
To enhance Modi character recognition, [11] proposed using a CNN autoencoder for feature extraction, reducing the feature set size from 3600 to 300, followed by SVM classification, which achieved a 99.3% accuracy, surpassing other methods.
Savitri Chandure and Vandana Inamdar [12] developed a supervised Transfer Learning (TL) based classification system, constructing a dataset for Modi handwritten characters. They utilized a pre-trained Deep Convolutional Neural Network (DCNN) AlexNet for feature extraction and SVM for classification, achieving recognition accuracy rates of 92.32% for Modi characters and 97.25% for Devanagari characters.

3. Structural Features of Modi Script

In the context of Modi script, the structural approach to character recognition, is insightful and aligns well with techniques commonly used for recognizing complex handwritten scripts. By focusing on the structural elements, such as strokes and their spatial relationships, this method offers resilience to variations in handwriting styles and distortions often found in historical manuscripts. Let me elaborate on how the components you mentioned can be applied effectively for Modi character recognition:

1. Contour Analysis

- **Modi Script**: Since Modi script is characterized by its flowing and connected nature, contour analysis helps identify the boundaries of each character or connected component. By analyzing the outer shape, the recognition system can distinguish between characters with similar internal features but different external shapes. This can also help in detecting ligatures or joined characters, which are common in handwritten Modi script.
- **Application**: Edge detection algorithms like Canny or Sobel can extract the contours, which can then be analyzed to derive the shapes of characters.

2. Stroke Decomposition

- **Modi Script**: Each character in Modi script can be viewed as a combination of various basic strokes like straight lines, curves, or hooks. Decomposing characters into individual strokes allows for better understanding and recognition. This decomposition can make it easier to handle character variations, as similar strokes will appear in multiple characters.

- **Application**: Techniques such as skeletonization or thinning can be used to reduce the character to its core structure, allowing for more accurate stroke identification. Following this, stroke classification techniques can group them into common sets of features.

3. Graph-Based Representation

- **Modi Script**: Characters can be represented as a graph, where the nodes are key structural points (such as intersections, endpoints, or corners), and edges represent the strokes or segments connecting them. This allows the system to capture the spatial arrangement and relationships between different parts of the character.
- **Application**: Graph matching algorithms can be used to compare the structural graph of an unknown character to known templates, facilitating more accurate recognition. This method is also tolerant of minor variations in stroke positions, making it robust against differences in handwriting.

4. Handling Document Degradation and Variability:

- **Modi Script**: Historical documents often suffer from ink smudging, fading, or other forms of degradation. The structural approach, by focusing on geometric and relational features rather than pixel-level information, is inherently better suited for handling such inconsistencies. It can distinguish between relevant character features and noise introduced by document damage.

5. Accommodating Handwriting Styles:

- Since Modi characters are often written in a connected manner and exhibit stylistic variations, a structural approach can generalize better across different handwriting styles by focusing on the core structural features, rather than pixel values which vary significantly with different handwriting.

This approach is especially powerful for recognizing ancient scripts like Modi, where the challenges of degradation and variability are pronounced. Do you have specific examples or datasets of Modi script that you would like to apply this approach to? We could also discuss suitable algorithms to implement each of these steps effectively.

Fig 1. Consonants in Modi Scripts

Fig 2. Vowels in Modi Script

4.Feature Extraction Techniques

Feature extraction from ancient Modi manuscripts presents unique challenges due to the frequently degraded condition of the documents, variability in handwriting, and the intricate structure of the script. To address these challenges, suitable methods for feature extraction should be considered, along with pre-processing steps to enhance image quality and reduce noise.

- **Pre-Processing for Image Enhancement**

Pre-processing for image enhancement is essential when working with ancient manuscripts, particularly for complex scripts like Modi. Here are some commonly used techniques for improving the quality of such images:

- **Noise Reduction**:Methods like Gaussian, Median, and Bilateral filtering can be used to reduce noise while preserving crucial structural details, such as edges and strokes.
- **Binarization**:Adaptive thresholding techniques, such as Otsu's thresholding or local methods like Niblack and Sauvola, can convert manuscript images into binary form. This process simplifies the image structure, making feature extraction more effective.

- **Contrast Enhancement**:**Histogram equalization** or **contrast-limited adaptive histogram equalization (CLAHE)** can be used to improve contrast and enhance faint features, making structural elements more distinguishable.

- **Feature Extraction Methods**

Feature extraction is essential in character recognition, particularly for complex scripts like the ancient Modi script. This process involves isolating and capturing significant features from images to streamline recognition while maintaining the script's core attributes. Ancient manuscripts present unique challenges, including degradation, diverse handwriting styles, and intricate character structures. To address these challenges, effective feature extraction techniques are vital for accurately identifying distinct shapes, strokes, and patterns. Methods such as Histogram of Oriented Gradients (HOG), Scale-Invariant Feature Transform (SIFT), and Zernike Moments provide robust tools for analyzing and distinguishing characters. These techniques improve character recognition in noisy or deteriorated conditions and accommodate variations in size, orientation, and style commonly found in historical documents.

- **Histogram of Oriented Gradients (HOG)**

Histogram of Oriented Gradients (HOG) is a powerful technique for feature extraction, particularly useful for character recognition tasks. It captures edge or gradient structures that are characteristic of the shapes within an image. HOG works and its application in character recognition, specifically for Modi script:

To compute HOG, the first step is to find the gradients in the image. The gradients emphasize the edges and highlight the structural features of the image. The gradient at a pixel $(x, y)$is computed using the following equations:

$Gy = I(x, y + 1) - I(x, y - 1)$

where,

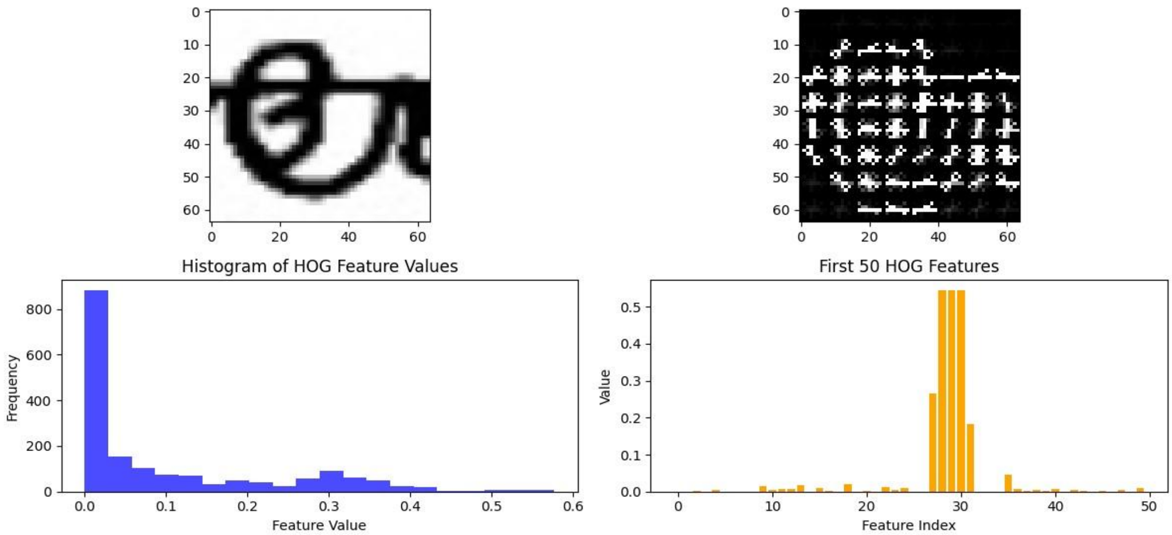
- Gx and Gy represent the gradients in the horizontal and vertical directions, respectively.
- $I(x, y)$ is the pixel intensity at location (x, y).

Using the horizontal and vertical gradients(Gx and Gy) , the magnitude $M(x, y)$and orientation $\theta(x, y)$of the gradient are calculated.

In the case of the Modi script:

- **Noise Reduction**: After applying noise reduction and binarization to enhance the quality of the document images, HOG is applied to extract features.
- **Stroke Analysis**: HOG analyzes the directional strokes, capturing the essence of Modi characters, which is vital for distinguishing between subtle variations in the handwriting of different individuals or degraded manuscript conditions.

HOG is a feature extraction method that focuses on the structure and shape of characters by capturing the gradients and edges. In the context of the Modi script, this method proves advantageous due to its robustness to variations in handwriting and its focus on local patterns, aiding in accurate recognition of the script's distinctive shapes.



Image(a)

In the images (a)above, each character is displayed with a set of plots that includes:

- **The Original Character Image**: The raw image of the character.
- **HOG Features Visualization**: An image showing the gradient orientations of the character.
- **Histogram of HOG Features**: A histogram representing the distribution of HOG feature values for the specific character.
- **Bar Plot of HOG Feature Values**: A bar plot displaying the first 50 HOG feature values for the character.

Each character's set of plots is processed individually, so the histogram directly reflects the features extracted from that single character. The bar plot visually depicts the magnitude of these 50 features, where taller bars represent stronger gradients in specific directions or cells, and shorter bars indicate weaker gradients.

- **Convolutional Neural Network(CNN)**

Convolutional Neural Networks (CNNs) have become a cornerstone in text recognition due to their ability to automatically learn and extract meaningful features from text images. Based on the analysis, Convolutional Neural Networks (CNNs) excel at image-based tasks like character recognition by automatically learning hierarchical features from input images. This capability of CNNs is particularly effective for complex scripts such as Modi, which involve intricate details and diacritical marks. In this context, the input consists of a binary image of the character, represented as a matrix of pixel values.

Convolutional layers use multiple filters on the input image to generate various feature maps. Each filter is designed to detect specific patterns such as edges, textures, or other local features within the image. In the context of text recognition, these patterns may include the various strokes and curves that make up the characters.

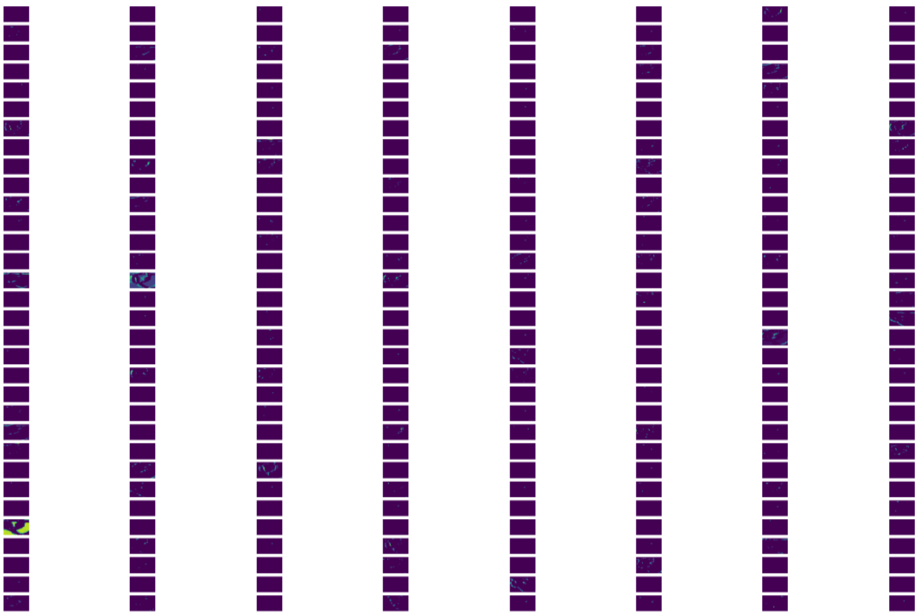
The convolution operation between an input image I and a filter Fcan be expressed as:

$$(I * F)(x, y) = \sum_{i=0}^{k-1} \sum_{j=0}^{k-1} I(x + i, y + j) \cdot F(i, j)$$

where:

- $I(x, y)$ is the pixel value at position (x, y) in the input image.
- $F(i, j)$ is the filter value at position (i, j) in the kernel.
- k is the size of the filter

Convolutional layers use multiple filters on the input image to generate several feature maps. Each filter identifies distinct patterns, such as edges and textures, within the image. In text recognition, these patterns capture the various strokes and curves of the characters.



Image(b)

In the images (a) above, character is displayed that includesthe internal workings of a convolutional neural network by displaying the feature maps of a specific convolutional layer. These feature mapsillustrate the activations of the various filters, revealing what the CNN has learned to identify, such as edges, textures, and patterns, from the input image at this stage. This visualization aids in understanding which features the CNN emphasizes during its predictions.

RESULT & DISCUSSION

- **HOG+KNN(k Nearest Neighbours)**

k-Nearest Neighbors (k-NN) is a non-parametric, instance-based learning algorithm utilized for classification tasks. The fundamental concept of k-NN is to assign a class to a data point based on the majority class among its nearest neighbors in the feature space. In this project, after extracting features from Modi characters using Histogram of Oriented Gradients (HOG), k-NN was used to classify the characters into their respective categories. The character images were processed with the HOG method, which captured crucial edge and texture information. These features were then fed into the k-NN algorithm. The optimal number of neighbors (k) was determined through cross-validation and grid search to achieve the best classification performance. The k-NN classifier was evaluated on a separate validation dataset, where it achieved an accuracy of 80%.

- **HOG+SVM**

Support Vector Machine (SVM) is a robust supervised learning algorithm employed for classification tasks. SVM functions by identifying the optimal hyperplane that best divides the data into distinct classes, with the goal of maximizing the margin between these classes. This approach is particularly effective in high-dimensional spaces, ensuring strong classification performance. In this project, SVM was utilized as an alternative to k-NN for classifying Modi characters, leveraging features extracted through the Histogram of Oriented Gradients (HOG) method. Initially, a linear kernel was used for the SVM classifier, followed by additional tuning with various kernels to capture potential non-linear relationships. The SVM model achieved an accuracy of 66% on the validation dataset.

- **Convolutional Neural Network(CNN for classification)**

After the convolution operation, the ReLU function is applied elementwise to the feature maps, activating the neurons and keeping only positive values. This process helps the network to learn detailed and intricate features of text characters.ReLU adds non-linearity to the model, enabling CNNs to capture complex patterns.

Pooling layers (such as max pooling or average pooling) reduce the size of the feature maps, decreasing the computational burden while retaining the most significant features. In text recognition, pooling aids in handling variations in character sizes and maintaining crucial structural elements.

Flattening converts the feature maps into a format that can be processed by the fully connected layers. This process transforms the spatial hierarchies learned by the convolutional layers into a one-dimensional vector, making it suitable for classification tasks.

Fully connected layers, also known as dense layers, utilize the features extracted by the convolutional and pooling layers to carry out classification or regression tasks. These layers combine the extracted features to generate final predictions, such as recognizing a specific character or categorizing different types of characters.The CNN model achieved an accuracy of 60% on the validation dataset.

CONCLUSION

The comparative study of feature extraction techniques for handwritten ancient Modi script recognition reveals several important insights. Techniques like Histogram of Oriented Gradients (HOG), and deep learning-based methods such as Convolutional Neural Networks (CNNs) each have distinct strengths and weaknesses. Traditional methods like HOG iseffective in capturing the fundamental structural features of the Modi script, particularly when working with smaller datasets. HOG excels at highlighting the intricate curves and edges characteristic of the Modi script, aiding in consistent recognition.

As the complexity of the script and the size of the datasets increase, deep learning methods like CNNs tend to surpass traditional approaches by automatically learning more abstract and high-level features. CNNs can identify complex patterns and variations in handwriting, making them well-suited for the intricate nature of ancient scripts. However, CNNs typically require a large amount of labeled data and significant computational resources, which can be challenging when dealing with ancient scripts that often have limited annotated datasets.

In conclusion, while traditional feature extraction methods offer a solid baseline for Modi script recognition, incorporating deep learning techniques, either alone or in conjunction with classical methods, can substantially improve recognition accuracy. The selection of a feature extraction method should be informed by the specific requirements of the task, including data availability, desired accuracy, and computational constraints.

REFERENCES

[1] Anil K. Jain, Template-based online character recognition, Pattern Recognition, Volume 34, Issue 1, January 2001, Pages 1–14

[2] A S Ramteke, G S Katkar, Recognition of Off-line Modi Script : A Structure Similarity Approach, “International Journal of ICT and Management” ISSN No. 2026-6839, February 2013

[3] D. N. Besekar, A S Ramteke, “Theoretical analysis of MODI script according to recognition point of view, some issues involved with character recognition of MODI script”, International Journal of Computer Applications, February 2013

[4] D. N. Besekar, “Recognition Of Numerals Of Modi Script Using Morphological Approach”,Shodh Samiksha Aur MulyankanVol.III, Issue-27, april 2011

[5] https://www.accusoft.com/resources/blog/ocr-vs-icr-whats-the-difference/ Published: March 23, 2021 (Updated: March 26, 2021).

[6] D. N. Besekar, A S Ramteke, “Chain Code Approach For Recognizing Modi Numerals , Indian Journal Of Applied Research”, December 2011.

[7] Algorithm and Kohenen Neural Network," International Journal of Computer Applications (0975 – 8887) Volume 111 – No 2, February 2015.

[8] Ramteke A.S., Katkar G.S., 2012, "Recognition of Offline MODI Script," International Journal of Research in Engineering, IT and Social

[9] Besekar D.N., Ramteke R.J., 2012, "Feature Extraction Algorithm for Handwritten Numerals Recognition of MODI Script using Zoning-based Approach," International Journal of Systems, Algorithms & Applications, Volume 2, Issue ICRASE12, ISSN 2277 2677, pp. 1-4.

[10] Besekar D.N., Ramteke R.J, 2013, “Study for Theoretical Analysis of Handwritten MODI Script – A Recognition Perspective,” International Journal of Computer Applications, vol. 64, no. 3, ISSN 0975-8887, pp. 45-49.

[11] S. Joseph and J. George, "Handwritten Character Recognition of MODI Script using Convolutional Neural Network Based Feature Extraction Method and Support Vector Machine Classifier," 2020 IEEE 5th International Conference on Signal and Image Processing (ICSIP), 2020, pp. 32-36.

[12] S. Chandure and V. Inamdar, "Handwritten Modi character recognition using transfer learning with discriminant feature analysis," IETE Journal of Research, pp. 1–11, 2021.

[13] Manisha s. Deshmukh, Manoj Patil, and Satish Kolhe," Offline handwritten Modi numerals recognition using chain code." WCI 15.

[14] Ankit Kumar Sahɔ, ShowmikBhowmikŞ, Samir Malakarș, Ram SarkarȘErginaKavallieratouȚ, Nikos Vasilopoulos “Text and Non-text Recognition using modified HOG descriptor”,IEEE Xplore,05 Feb 2018