Gujarati Optical Character Recognition Using Efficient Text Feature Extraction Approaches

Avani Samir Bhuva, Dhirendra Mishra
Computer, MPSTME (NMIMS) Bhakti Vedant Marg, opp. Cooper Hospital, Navpada, Suvarna Nagar, Vile Parle West, Mumbai, India
E-mail: savani751986@gmail.com, dhirendra.mishra@nmims.edu

Keywords: image processing, optical character recognition, gujarati text feature vector dictionary, integrated shape numeric encoding approach (ISNEA), one-bit frequency count approach (OBFCA), fusion of region geometric features (FRGF), CNN, RNN

Received: February 27, 2025

India is the most populous country, with 22 official regional languages. Retrieving information from these regional languages is a challenging task. Approximately 62 million people worldwide speak the Gujarati language. This research paper aims to understand and extract the meaningful text features of the Gujarati text from the OCR Gujarati dataset. This research focuses on extracting meaningful text features from the Gujarati OCR dataset, which comprises 23,100 samples generated using the TERAFONT-VARUN font and augmented with horizontal/vertical shifts and rotational transformations. This study explores three levels of text feature extraction: Mid-level features using the Integrated Shape Numeric Encoding Approach (ISNEA) and Fusion of Region Geometric Features (FRGF), Mid-high-level features via the One-Bit Frequency Count Approach (OBFCA), and High-level features through a deep learning-based CNN model. The extracted features were stored in a structured Gujarati Text Feature Vector Dictionary. ISNEA struggles with characters containing maatra's modifiers, with 87.5% on standardized OCR images. OBFCA resolves the maatra's issue by row-wise binary frequency computation, yielding 90.11% accuracy. FRGF significantly outperforms ISNEA and OBFCA, with 93.5% accuracy using Eccentricity as a single feature and 92.75% using Eccentricity + Perimeter as a fused feature. The Euclidean distance and cosine similarities were also used to measure the similarities between the extracted text features. Comparative analysis against existing methods confirms the superiority and robustness of the proposed approaches in Gujarati OCR feature extraction.

Povzetek: Predstavljene so tri metode za učinkovito ekstrakcijo značilk iz slik z besedilom v gudžaratiju (ISNEA, OBFCA, FRGF), pri čemer geometrijske značilke dosegajo najvišjo točnost.

1 Introduction

The utilization of the internet has significantly evolved over the past decade. Users typically formulate queries to seek information online, subsequently receiving relevant explanations. With the advancement of digital technology, data can be effortlessly accessed from various formats, including documents, images, and videos uploaded daily. Recent surveys show over 150,000 new videos are added to YouTube every minute. After the COVID-19 pandemic, approximately 3.7 million videos are uploaded daily, amounting to around 271,330 hours of video content. This digitized information is available in multiple formats. It varies in size across diverse domains such as education, cooking, sports, and entertainment, as well as in numerous languages, including English, Hindi, Gujarati, Tamil, and others. However, extracting information from these digitized resources presents a considerable challenge. Converting documents or images into machine-readable text is called optical character recognition (OCR). An OCR system interprets images containing printed or handwritten characters and converts them into editable text formats. This involves segmenting the text image into lines and subsequently into individual characters, creating distinct areas for analysis. During character extraction, various attributes of the text image are evaluated, including corner points, features of different regions, the ratio of character areas, and the convex area encompassing all characters. The OCR system's workflow is depicted in the accompanying diagram. It starts with dataset collection and then applies different image processing algorithms to the images for pre-processing and feature extraction. Computer algorithms are used in digital image processing to handle massive amounts of documents, videos, and photos. Meaningful information can be extracted from these digital datasets by applying image processing techniques. Since the information comes in different forms, specific techniques are required for effective extraction. This study looks at a number of image processing methods for textual content information extraction. Gujarati is an Indo-Aryan language that originates back to the tenth and eleventh centuries. Originating in Gujarat, India, it has evolved into a dynamic language inspired by Arabic, Persian, English, Sanskrit, Prakrit, and Apbhransh. The three most commonly spoken languages among the 7,139 recognized worldwide are Hindi, Chinese, and English. After Papua

New Guinea, India has the second-highest linguistic diversity.

After Hindi, Gujarati is the sixth most spoken language in IndiaThe complexity of Gujarati is demonstrated by the knowledge that the same words can have distinct meanings depending on the gender of the speaker. Gujarati nouns are classified as feminine, masculine, or neuter, and they can be either singular or plural. For example, "bahen" (બહેન) means "sister" in the feminine, and "bhāī" (ભાઈ) means "brother" in the masculine. Furthermore, nouns like cities and lakes are neutral, rivers are feminine, and countries and oceans are usually masculine. As shown in the accompanying figure, the language consists of 36 consonants, 26 vowels, and numerical symbols.

Figure 1: (a)-(b) represents Gujarati Text (c) represent Gujarati number

Features in image processing are the distinctive traits or patterns found in an image that provide important details about its composition, appearance, and content. Tasks like object detection, recognition, and classification are made easier by the feature extraction process, which is crucial for the practical analysis and interpretation of images. Features can be classified as either high-level, which represent more abstract concepts like shapes or areas of interest, or low-level, which capture basic characteristics like edges and textures. An image can be mathematically expressed as a two-dimensional function I(x,y), where x and y are the spatial coordinates (horizontal and vertical positions) of each pixel. The intensity or color value of each pixel, whether in RGB or grayscale format, is provided by the function I(x,y). In image processing and pattern recognition, extracting these features is a crucial step that transforms unprocessed data into a set of valuable characteristics, or "features." These features, which can be used for classification, clustering, and detection tasks, capture important information from the data, such as shapes, textures, colors, or patterns. By lowering the dimensionality of the data while retaining its fundamental characteristics, feature extraction simplifies analysis. Mathematically technically this process is a transformation that turns the input data, X, into a feature space, ϕ(X), where each dimension denotes a unique feature:

$$\phi(X) = \{f1(X), f2(X), \ldots, fn(X)\} \tag{1}$$

The original image is represented by X in this equation, the transformed data is represented by ϕ(X) in the feature space, and the ith feature extracted from X is indicated by fi(X). Following feature extraction, we commonly combine or sum individual feature values to create a final feature score or composite feature vector when several features are combined to represent a data point. The final feature sum can be calculated using the following general formula, where each feature adds to the cumulative score:

Let fi(x) stand for the ith feature that was taken from the data X, and let ω_(i) be the weight that is given to the fi(x), indicating how important each feature is to the final score. Then, the final feature score *F(X)* is given by the weighted sum of all features:

$$F(X) = \sum_{i=1}^{n} \omega i \, f_i(x) \tag{2}$$

Where n is the total number of features, $\omega_i f_i(x)$is the weighted contribution of the i^th feature of the final score. *F(X)* represents the aggregate or cumulative feature score for *X*.

In image processing, text feature extraction is a specialized technique used to locate, identify, and extract textual information from images. Applications involving image-based language processing, scene text recognition, and document digitization all depend on this method. Computers are able to understand and analyze text embedded in images through the process of extracting textual features. Applications like automated document analysis, natural language processing (NLP), and optical character recognition (OCR) depend on this feature. Images are examined to find patterns that correspond to specific words or characters during text feature extraction, and these patterns are then transformed into structured text data. Different fonts, backgrounds, orientations, and noise that could mask the characters are some of the unique challenges that text feature extraction poses in contrast to traditional object recognition. Below is a summary of several text feature extraction techniques.

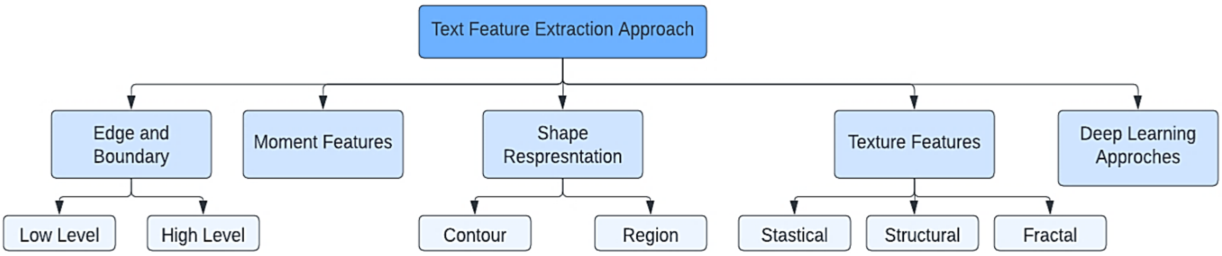


Figure 2: Classification of text feature extraction approaches

The organization of the paper is as follows: Section 2 provides a review of related work concerning Gujarati text feature extraction techniques. Section 3 details the proposed methodologies. Section 4 showcases experimental results along with a comparative analysis. Finally, Section 5 concludes with a summary of the proposed method and discusses avenues for future research.

2 Related work

Since COVID, the amount of digital documents has grown, making it difficult to accurately extract relevant information from them. These features allow for the analysis, interpretation, and extraction of pertinent information by capturing details about important aspects of the image. Features are crucial for image analysis tasks such as object detection, image recognition, segmentation, and others. In image processing, the term "text features" refers to traits or qualities associated with the text contained in images. Text features, which are crucial for optical character recognition (OCR), document analysis, scene text recognition, and many other applications, are the main focus of this study.

Many studies have focused on handwritten digits, isolated characters, and full text recognition when developing optical character recognition (OCR) systems tailored for Gujarati text. Early research in this area primarily utilized conventional image processing and feature extraction methods like edge detection and template matching. These techniques performed well with inputs that were clear-cut and well-structured, but they had trouble with complicated font patterns and variations in handwriting styles.

To address the different issues involved in text recognition from scanned documents, images, or other sources, there are several types of text features, such as OCR Features, Text Region Features, Text Texture Features, Text Layout Features, and Text Color Features. In image processing, there are numerous methods for extracting text features, including Bounding Box Coordinates, Skeletonization, Text Localization, and Connected Components. In-depth research on various Deep Learning techniques for text feature extraction has been presented by Hong Liang et al. [3]. Several image processing-based feature extraction techniques have been covered by S.A. Rajesh et al. [4], [5], and A. O. Salau et al. [33]. Gujarati text processing is difficult because of its complex morphology.

The rich morphology of Gujarati, which results in a single word having multiple versions, makes it difficult for the Gujarati language, according to researchers Nikita P. Desai et al. [6]. In order to recognize Gujarati handwritten digits, Anardan Bharvad et al. [7] have compared a number of techniques, such as the Stroke Orientation Estimation Technique, Neural Network, Naive Bayes, Sparse Representation Classifier, Low-level Strokes, Support Vector Machine, and Wavelet Features Extraction. Despite considering the gradient's direction as a function vector, the researcher's accuracy was noticeably lower. Several methods for retrieving documents in Gujarati text were covered by S. Gautam et al. [8].

By contrasting various text recognition, feature extraction, and feature matching techniques—all of which can be applied to retrieval approaches—this research paper has also produced a thorough literature review. Different Gujarati text recognition techniques, Gujarati script properties, and the recognition of printed and handwritten documents were all covered by K. B. Khushali et al. [9]. Using the skeletonization concept, the researchers Jyoti Pareek et al. [10] have put forth a novel technique for obtaining features from unprocessed character images. The researcher [11] suggested an OCR method for Gujarati handwritten digits and found that the recognition rate of Gujarati handwritten digits is also low when compared to Kannada, tamil, telugu and that the verification rate for numbers like zero, four, and seven is very high (98%), and for digit six, 72%. An overview of the research paper's literature is provided in Table 1, which also includes information on the datasets used, feature extraction techniques used, and accuracy attained.

The recognition of Gujarati handwritten text and numbers has been the subject of extensive research using a variety of datasets, sample sizes, and text feature extraction techniques. These studies' sample sizes range greatly, from as few as 22 samples to over 14,000 samples. Conventional methods, such as the Chain Code Histogram, Invariant Moments, and Matrix Pattern Method, have attained moderate accuracy levels, typically ranging from 80% to 91%. For example, the accuracy of the skeletonized matrix pattern method was 91.76%, whereas the accuracy of the Morphological Transformations plus K-NN was 82.03%. Deep learning techniques, on the other hand, have shown notable improvements in classification accuracy. On datasets with 10,000 and 14,000 samples, respectively, Convolutional Neural Networks (CNNs) have demonstrated remarkable accuracy rates of 97.21% and 99.81%. EfficientNet-based models have also demonstrated impressive performance; one study used datasets from newspaper articles and achieved 96.5% accuracy, while another study achieved 99.70%. On the other hand, Gujarati text recognition has been less successful with object detection models like YOLO, EfficientNet, and Faster R-CNN, with accuracies ranging from 42.64% to 60%.

High performance, including a 97% accuracy rate on a dataset of 6,000 samples, has been reported in studies employing LSTM and other deep neural networks. However, some techniques produced lower accuracies of roughly 67.95%, such as those that used HoG features with Convolutional Autoencoders. In conclusion, the findings show that Gujarati OCR systems' accuracy is greatly increased when deep learning architectures are combined with larger datasets. The table below provides a summary of the thorough literature review.

Table 1: Literature survey on gujarati text feature extraction approaches

Paper	Dataset	Sample size	Text Feature Extraction Approach	Classification Accuracy
[7]	Gujarati Handwritten Digits	600	Matrix Pattern Method	Without Skeleton 89.83%, with Skeleton 91.76%
[8]	Handwritten Data	9318	HoG and SD	Convolutional Autoencoder 67.95%
[9]	Gujarati Handwritten Text	22	Histogram of Oriented Gradient, Chain Code Histogram	SVM -88.4%
[10]	Gujarati Handwritten Text	10000	--	CNN- 97.21% 64.28%
[11]	Gujarati Handwritten Digits	300	Pattern Matrix	Feed Forward Back Propagation Neural Network- 81.66%
[12]	Gujarati Handwritten Digits	5000	Invariant Moments	Neural Network 80.5%
[13]	Gujarati Handwritten Digits	600	Block wise division character	Naive Bayesian model 80.5%
[14]	Gujarati Handwritten Digits	2500	Pre trained CNN networks	EfficientNet 96.5%
[15]	Gujarati Handwritten Digits	14000	--	CNN- 99.81%
[16]	Newspaper, articles etc.	500	Darknet-53	FCN
[17]	Gujarati Characters	6000	Deep Learning	LSTM-97.00%
[18]	Gujarati Handwritten Text	--	DNN	Neural Network
[19]	Gujarati Handwritten Text	30	Morphological Transformation	K-NN 82.03%

[20]	Gujarati Handwritten Text	200	--	Efficient Det-60% YOLO-42.64% FASTER RCNN - 55%
[21]	Gujarati Newspapers Articles	6558	--	68.17% EfficientNetB3-99.70%
[22]	Gujarati Handwritten Text	5980	--	CNN- 94.8%
[23]	Online image from internet	--	--	k-Nearest Neighbor 67.00% The Minimum Hamming Distance 39%

Algirdas L. et al. [43] present a novel semi-automated technique for class label confirmation and modification in multi-label text datasets, which employs self-organizing maps (SOM) to detect class similarities and latent semantic analysis (LSA) to reduce dimensionality. By identifying and correcting incorrect class assignments, cosine similarity improves the quality of the data. Using a manually classified Lithuanian financial news dataset with 10 classes, the technique achieved an accuracy rate of 82% incorrect class assignments, demonstrating that it significantly improves classification accuracy and helps to produce higher-quality datasets for further data analysis tasks.

In order to address the low precision of current alignment algorithms in literary texts, Pavel S. et al.'s study [44] presents a novel framework for extracting translation memory from a bilingual corpus of fiction and non-fiction books. The suggested method greatly improves alignment accuracy by combining conventional alignment techniques with a proactive learning strategy and creating feature functions for training two classifiers: one for alignment and one for text filtering. The effectiveness of the framework in enhancing translation memory extraction for intricate, non-technical texts is demonstrated by experiments conducted on a corpus of 200 English-Lithuanian books, which reveal significant improvements over current systems. This section and this thorough literature review have covered a variety of languages. A method for identifying words in grayscale Pashto documents written in modified Arabic scripts was presented by Ismail Shah et al. [24]. Grayscale images' features are taken out and transformed into binary feature vectors. The data set is 4200 words in size. With an average recall of 60.25%, an average precision rate of 94.75% was attained. Kolcz et al. [25] suggested a line-oriented technique for extracting text in Spanish from a document that was 13 pages long. Enver Akbacak et al. [26] used dynamic time warping to extract a single feature from 15 images containing 2381 English words. Additionally, Yogya Tewari et al. [27] have created a model that converts signs from American Sign Language (ASL) into English letters by using CNN to split the signs into the appropriate English alphabet. The model's accuracy in recognizing 26 alphabets and three additional characters reached 99.78%. The chain code method has been used for OCR in handwritten Arabic by Hassan Althobaitim et al. [30]. Based on character input, the accuracy ranged from 92% to 97%. Muhammad Arif Mohamad et al. [31] used the Chain Code technique to extract the features from the Document Analysis and Recognition (CEDAR) dataset. They reported that this method takes 1.10 seconds to solve the full set of character images.

Numerous Indian languages, including Hindi, Kannada, Tamil, Bangla, Malayalam, and Gurumukhi, have undergone extensive research, according to researchers Mohamed Fakir et al. [28]. English makes up 38% of the 7000 recognized languages in the world, followed by Chinese at 33% and other regional languages at 2%. According to the literature, only 2% of research has been completed on regional languages like "Gujarati," which is spoken by over 66 million people and is the 26th most widely spoken language in the world. From the literature review, we have determined the following challenges.

2.1 Challenges in gujarati OCR

1. The lack of comprehensive and diverse datasets covering a range of document formats, fonts, and handwriting styles.

2. The majority of currently available datasets have a limited scope, frequently focusing on handwritten numbers, small character sets, or discrete printed text collections.

3. Overfitting, in which models perform well on training data but find it difficult to adjust to real-world applications, is caused by this lack of diversity in datasets.

Furthermore, a lot of existing approaches fall short in capturing the complex geometric features or regional subtleties of Gujarati characters, which hinders their efficacy when working with complex or noisy input data.

2.2 Research questions

This study examines optical character recognition (OCR) for the Gujarati language with the goal of increasing accuracy and efficiency given the script's intricacy and the dataset's scarcity. The following research questions have been developed to guide the investigation:

RQ1: How can text feature extraction techniques be optimized to increase the accuracy of OCR systems for the Gujarati language?

RQ2: Can Gujarati characters and numerals be identified more accurately using a suggested model that incorporates shape-, frequency-, and region-based geometric features than current techniques?

RQ3: What is the impact of feature-level fusion integration on Gujarati text recognition systems' classification accuracy.

2.3 Research hypotheses

To address the above questions, the study proposes the following hypotheses:

H1: The Integrated Shape Numeric Encoding Approach (ISNEA), One-Bit Frequency Count Approach (OBFCA), and Fusion of Region Geometric Features (FRGF) are anticipated to outperform the conventional feature extraction methods currently used in Gujarati OCR in terms of classification accuracy.

H2: Among these three methods, Due to its ability to capture intricate geometric and spatial features across multiple regions, the FRGF is predicted to produce the highest recognition accuracy among these three techniques.

H3: The development of a structured Gujarati Text Feature Vector Dictionary is expected to enhance classification efficiency and improve recognition performance, even when working with a moderately sized dataset

2.4 Motivation and problem definition

1. Gujarati optical character recognition (OCR) using Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and LSTM architectures has made significant strides, but the top techniques still have a number of issues. One significant problem—which will be covered in the next section—is that many of these models depend on large, evenly distributed datasets in order to achieve high accuracy levels. The large and varied datasets required, particularly those that capture the complex structure of handwritten characters and the stylistic variances found in real-world documents like newspapers or digitised texts.

2. Additionally, most current techniques concentrate on either spatial (image-related) or sequential (language-related) features in isolation, failing to combine visual structure with linguistic context effectively. Primarily ignoring exhaustive geometric and regional text features, methods of feature extraction—including Histogram of Orientated Gradients (HoG), Invariant Moments, and pre-trained Convolutional Neural Networks (CNNs)—produce poor results under overlapping characters, degraded text, or font style variation.

3. Our proposed solution addresses these shortcomings by presenting a framework that extracts distinct text feature extraction techniques, ISNEA, OBFCA, and FRGF, each aimed at capturing unique structural, frequency-based, and geometric features of Gujarati characters. Notably, the Fusion of Region Geometric Features (FRGF) enhances the ability to differentiate by examining various spatial regions, making it particularly adept at managing intricate text structures. Developing a Text Feature in Gujarati Effective classification becomes possible by Vector Dictionary, even in situations where data is limited. With a remarkable accuracy of 93.50% with eccentricity-based features, this comprehensive approach not only enhances recognition accuracy but also fosters better generalisation across different input styles, outperforming many current benchmarks.

The problems listed below would be resolved by creating a Gujarati-specific text feature extraction approach that makes use of current techniques like deep learning and image processing algorithms to manage complex scripts, noisy data, and a variety of text styles. This approach would enhance Gujarati text recognition systems' accuracy and advance the more general objective of allowing the digital age accessibility and usability of regional language data. Numerous research gaps have been identified from the literature [7]-[23] regarding text feature extraction methodologies over Deep Learning and Image processing concepts. The literature has revealed certain research gaps are outlined in the table below

Table 2: Detailed analysis of research gaps concerning deep learning approach

Research Gap	Underlying Issues	Proposed Solution
Dataset utilization [7]-[23]	Existing major research work, as mentioned in [7]-[23] done under the Gujarati digits, is limited to Gujarati text.	Worked over Gujarati OCR text dataset with 23100 samples of text variations
Lack of Explainability [14]-[22]	Deep learning models function like black boxes, which makes it hard to grasp how they pull out features and what causes specific mistakes.	Incorporating explainability techniques, we have provided our proposed solution with detailed text feature values and presented them in the table and in a graphical manner.
Computational Efficiency [14]-[22]	Deep learning models for text feature extraction are often resource-intensive, making them impractical for deployment on edge devices.	Image processing models don’t require high computational devices for the feature extraction process

Generalization Across Languages [24]-[28],[30]-[31]	Existing Deep models are often trained on datasets from high-resource languages (like English) and may not generalize well to low-resource or complex scripts (e.g., Gujarati, Devanagari, Arabic, etc.).	Creating Gujarati language-agnostic text feature extraction frameworks that adapt to various scripts and styles.
Complex Fonts and Handwritten Text [7]-[23]	Extracting features from Gujarati handwritten or stylized text remains challenging due to the variability in character shapes, spacing, and strokes.	Developed image processing approaches that can identify the 60 augmented variations of a single alphabet from 385 classes.
Text Feature Extraction for Non-Latin Scripts [24]-[28],[30]-[31]	A lack of publicly available datasets and models optimized for non-Latin scripts (e.g., Gujarati, Tamil, Chinese) limits research in this area.	Curating diverse, open-source datasets for Gujarati languages and designing script-specific feature extraction pipelines.
Multi-modal Feature Integration [7]-[28],[30]-[31]	Current text feature extraction methods focus only on single feature values, not composite feature values.	We have proposed the fusion approach, which successfully extracts the fusion features.

2.5 Unique contribution to underlying issues and challenges

1. We introduce an innovative Shape Numeric Encoding Approach (ISNEA) that creates a pixel-level representation of the shape of text images, emphasizing their boundaries. These numerical representations are compiled as features, which will be the unique length of each feature, resulting in a detailed feature vector integrated into the Gujarati text feature vector dictionary. This technique effectively captures the shape characteristics of the text in a numerical format.
2. In addition, we propose the One-Bit Frequency Count Approach (OBFCA), which aims to extract the central axis of the text by employing morphological operations, thereby producing a thin boundary representation. Once this thin boundary image is generated, it is converted into a character array to obtain the numerical values of the grayscale image. This grayscale character array is then transformed into a binary character array through pixel value substitution. In the concluding phase, we calculate the frequencies of 1's row-wise for a 32x32 image, with the cumulative frequency recorded as a feature vector in the Gujarati text feature vector dictionary.
3. We also proposed the third approach, in which we advance the framework for regional analysis by incorporating connected components to determine the number of regions within the text. We extract geometric features from these text regions, which are subsequently categorized into distinct and combined features.
4. We perform comparative experiments to evaluate the efficacy of the proposed framework utilizing the Gujarati OCR dataset, comparing it against existing and cutting-edge methodologies.

3 Proposed approaches for text feature extraction

3.1 Datasets description

The proposed uses the Gujarati OCR dataset introduced by Anand R et al. [34]. This dataset includes images rendered in the "Shruti" font style, featuring a font size of 12 and various stylistic variations tailored explicitly for the Gujarati language. The images are presented in RGB format. The collection encompasses 34 characters of the Gujarati language along with 11 variants of modifiers (Maatra's), resulting in 374 distinct characters and 11 vowels, which makes a total of 385 characters.

To ensure consistency, all images are standardized to 32x32 pixels. Additionally, the dataset undergoes image augmentation techniques, including horizontal and vertical shifts, zooming, rotation, and brightness adjustments, to create further variations for each character. The following table provides a detailed analysis of the dataset utilized for the three proposed approaches: ISNEA, OBFCA, and FRGF, both as individual features and in fusion.

Table 3**:** Description of the gujarat text dataset

Attribute	Details
Dataset Name	Gujarati OCR Dataset
Image Resolution	32 × 32 pixels
Font Style Used	TERAFONT-VARUN
Total number of training images	53,043
Total number of testing images	23,100
Total Number of Classes	385 (including consonants, vowels, and matras)
Character Breakdown	34 consonants, 11 independent vowels, 11 matras variations
Samples per Class	60 augmented samples per class
Data Augmentation Techniques	Horizontal & vertical shift, zooming, rotation, brightness adjustments

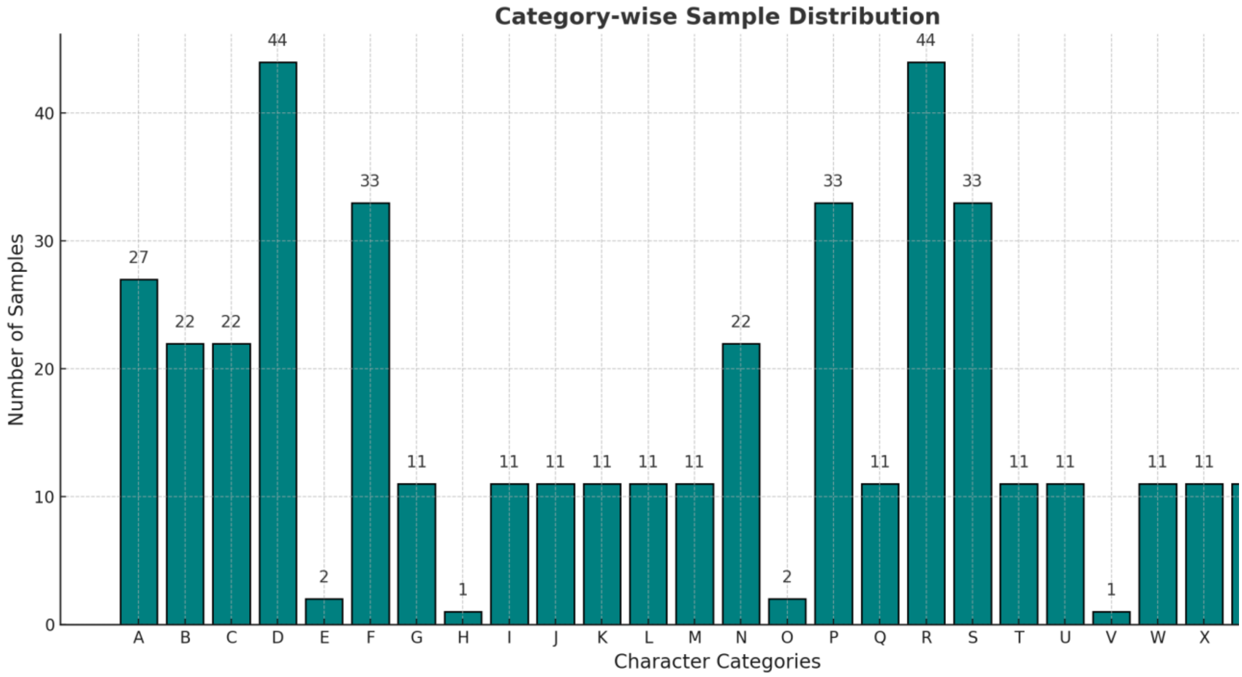


Figure 3: Category-wise sample distribution of gujarati OCR character dataset [34]

This research was undertaken to develop a feature vector dictionary. Our subsequent objective is to expand this investigation by integrating it with content obtained from video frames. After extracting Gujarati text features, these will be evaluated against our proposed Gujarati text feature vector dictionary. Additionally, training convolutional neural networks requires hardware support and considerable time investment to process the text features effectively. Currently, we have not employed all the state-of-the-art techniques other than CNN in this research endeavor. However, in the experimental section, we have compared the proposed methods with the CNN architecture. The basic architecture of the proposed system is shown in the image below.

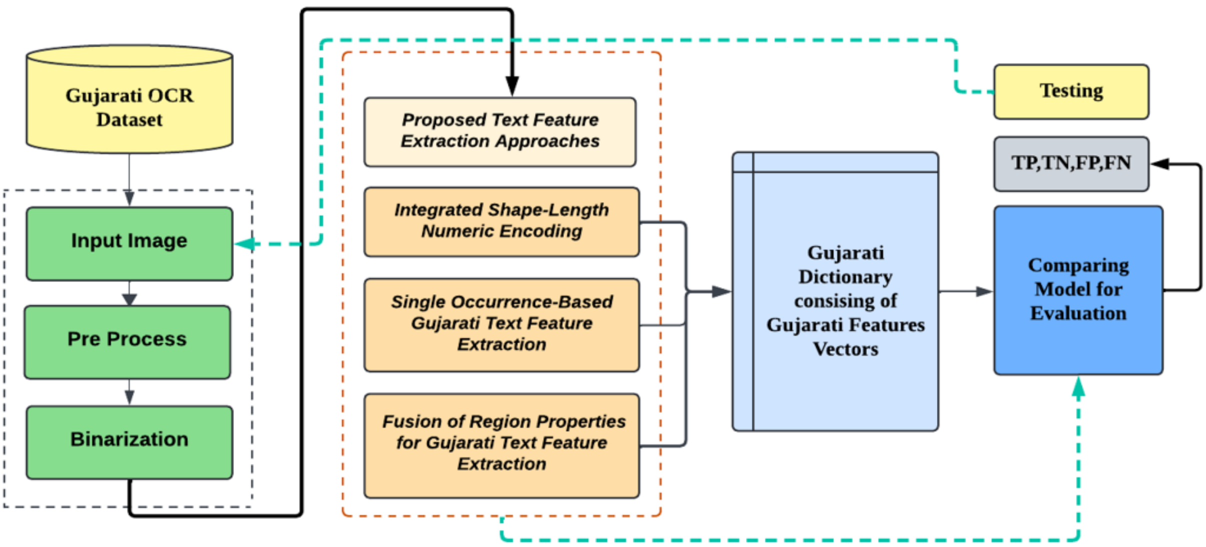


Figure 4: Proposed workflow for gujarati text feature extraction approach

The system's architecture is organized into three distinct stages: (i) Preprocessing and Binarization, (ii) Extraction of text features utilizing the proposed methodologies, and (iii) Storage of the extracted features in a Gujarati feature vector dictionary. In the first stage, input text images undergo a Binarization process. The outcomes of this initial stage are visually represented in Fig. 4. The second stage focuses on the extraction of shape, boundary, and geometric features specific to Gujarati text, with the results depicted in Figs. 5, 7, and 10. The numeric feature vectors are presented in Tables 4,5,6, and 7 for the ISNEA, OBFCA, FRGF approaches.

3.1 Integrated shape numeric encoding approach (ISNEA)

The morphological richness of the Gujarati language contributes to a high complexity in its textual structure. As a result, the tasks of processing and feature extraction are notably challenging. This section will address the first proposed approach, which aims to extract shape features from Gujarati text through a numerical encoding method. A detailed block diagram outlining the proposed methodology is displayed below.

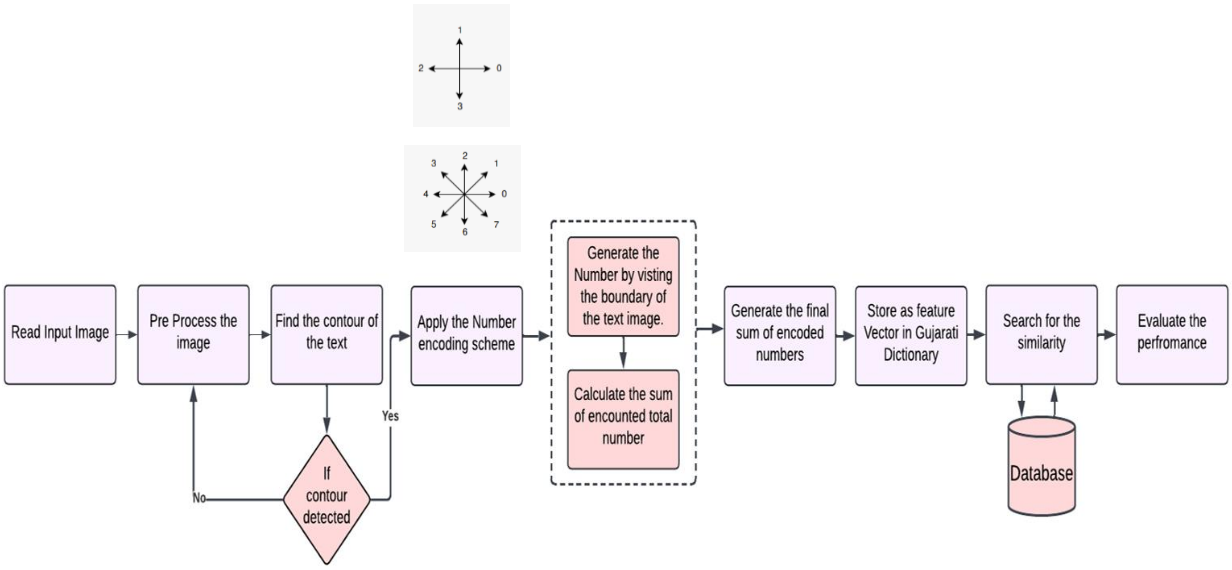
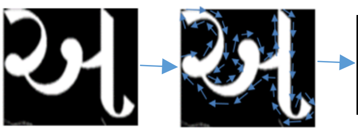


Figure 5: Proposed detailed diagram on integrated shape numeric encoding approach (ISNEA)



F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	F21	F22	F23	F24	F25	F26	F27	F28	F29	F30	L
0	5	7	6	7	1	0	5	6	7	6	7	0	7	0	2	3	2	1	3	1	7	1	1	4	5	6	4	5	4	30

Figure 6**:** Numeric Encoding Feature for Gujarati Alphabet ‘A’

As we can see in the above figure, by using the Number coding scheme, we have labeled the boundary of the text ‘Aa’ based on the arrow direction. The overall sum will be calculated as a feature vector and will be stored in the Gujarati Feature Vector dictionary. This entire process has been discussed in the algorithm below.

Algorithm 1 Integrated Shape Numeric Encoding Approach (ISNEA) to Extract Gujarati Text Features

Input: *Gujarati Text OCR Images*
Output: *Extract the Gujarati text features and generate the feature vector file.*
Procedure:

Step 1: Preprocess the Input Image
- Convert to grayscale if necessary
- Apply binary thresholding to separate foreground (text) from background.

Step 2: Extract Contours
2.1 Use the contour detection method to find boundary contours in the image
- If no contours are found:
  Go back to Step 1 (image preprocessing)
- Else, proceed to generate numeric encoding:
2.2 Select the first external contour:

$$C = \{(x_0,y_0), (x_1,y_1),.. (x_n, y_n)\} \quad (3)$$

For each point in the contour, compute numeric encoding $N_{ei} = \{ne_0, ne_1, ..., ne_{n-1}\}$ (4)

where n_{ci} = direction $((x_i, y_i), (x_{i+1}, y_{i+1}))$

Step 3: Compute Relative Position for Direction

Encoding

3.1 For each consecutive point pair (x_i, y_i) and (x_{i+1}, y_{i+1}):

$$\Delta_x = x_{i+1} - x_i \quad (5)$$

$$\Delta_y = y_{i+1} - y_i \quad (6)$$

Determine direction code di based on Δx and Δy using an 8- direction table

e.g., if Δx=1 and Δy=0, then direction = 0 (right)

if Δx=-1 and Δy=-1, then direction = 5 (down-left).

Traverse along the boundary of the text over the entire image

Step 4: Generate the Final Feature Vector and the final feature score as mentioned in **equ.(2)**

Step 5: Store Feature Vector

- Save the resulting vector in a Gujarati Dictionary File for future retrieval

Step 6: Match Feature Vector

- Compare with existing entries in the dictionary using Euclidean and cosine distance

- Retrieve the most similar match

Step 7: Calculate the performance of the proposed system

- precision, recall, and F1-score.

The numeric Encoding is presented in the table below, where NE3 is the numeric value for image 3, NE14 for image 14, and so on. Alongside L is the overall length of the features together, and stored in the Gujarati dictionary for the matching criteria.

Table 4: Distinct Text Features of ISNEA Approach (NE3= Numeric Encoding for image-3, L=length)

NE	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	F21	F22	L
NE3	7	7	0	6	7	7	5	5	6	5	6	4	4	5	1	1	1	0	3	3	2		85
NE14	7	0	7	0	0	6	7	5	5	6	4	1	3	4	3								58
NE29	7	7	7	0	7	0	6	7	7	5	5	6	5	6	4	5	2	1	1	0	4	3	95
NE58	1	2	0	1	7	0	7	5	5	6	4	4	5	3	4	3	4	3	2	3	2	3	74
NE60	7	0	7	0	7	0	6	7	7	5	5	6	5	6	4	2	1	1	0	3	3		82
NE70	7	7	7	1	7	0	6	7	7	5	5	6	5	7	4	2	1	1	1	3	3		92

The proposed ISNEA approach encounters difficulties in the following scenarios:

1. Ignore inner or secondary contours
2. Fail to incorporate positional relationships between regions
3. Produce a single-directional feature that may be insufficient for full shape representation.

To overcome these issues, we have proposed the new approaches OBFCA and FRGF which will be discussed in the following section.

3.2 One-Bit frequency count approach (OBFCA):

This method is dividing into three stages. Stage-I generating the thin boundary of the input image, stage-II generating the char array of the thin boundary image and stage-III store char array as feature vector in Guajarati Text dictionary. The proposed detailed diagram is shown in Fig. 7.

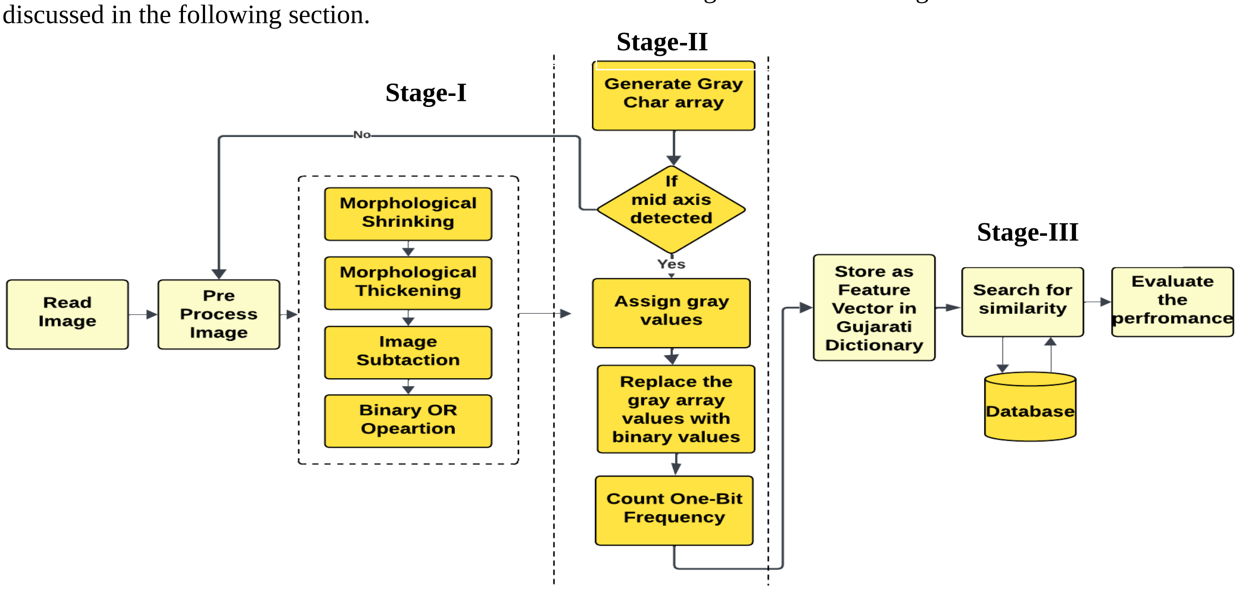


Figure 7: Proposed detailed diagram for one-bit frequency count approach (OBFCA)

The detailed explanation of the proposed approach is mentioned below.

***Algorithm 2** One-Bit Frequency Count Approach (OBFCA)*

Input: *Gujarati Text OCR Images*
Output: *Extract the Gujarati text features and generate the feature vector file.*
Procedure:
Step 1: Preprocessing
- Read a grayscale image
- Apply binary thresholding (invert background)
Step:2 Create a black mask the same size as the image
2.1 Apply iterative morphological thinning to generate a skeleton image
2.2 Save skeleton image.
Step:3 Convert Skeleton to Grayscale
3.1 Read the skeletonized image
3.1 Convert to grayscale
Step:4 Extract Gujarati Text Features
4.1 Find contours of the characters
4.2 Draw bounding boxes if necessary
4.3 Convert grayscale pixel matrix to DataFrame
4.3.1 Replace all pixel values: 255 ➔ 1 (foreground), 0 ➔ 0 (background).
Step 5: Save Feature Matrix
- Save the binary DataFrame to Excel file
Step 6: Post-process Feature Matrix
6.1 Replace 0 values with NaN if required
6.2 Count number of 1's (foreground pixels) in each row
6.3 Sum the counts to obtain final feature length
Step 7: Search in Reference Dataset
7.1 Load reference dataset from stored feature vectors
7.2 Match rows where 'length' matches the calculated feature sum
Step 8: Store Feature Vector
Save the resulting vector in a Gujarati Dictionary File for future retrieval
Step 9: Feature Vector Comparison
9.1 Calculate Euclidean distance between test image feature and reference vectors
9.2 Find the reference vector with minimum distance
Step 10: Generate the Output
10.1 Report the matched Gujarati character with the minimum distance
10.2 Report the minimum Euclidean distance value
Step:11 Calculate the performance of the proposed system
- precision, recall, and F1-score

In the proposed study, the generated character array will have all pixel values ranging from 0 to 255 replaced with 1, while the remaining values will be set to 0. This results in a binary character array. Upon zooming out of the character array, it becomes evident that Figure d illustrates the shape of the Gujarati text 'A' with a length of 61. This process has been applied to all 23,100 images, converting them from grayscale values to binary values to obtain the final count. Therefore, our proposed method effectively extracts the slender shape of the Gujarati text, as demonstrated in the image below.

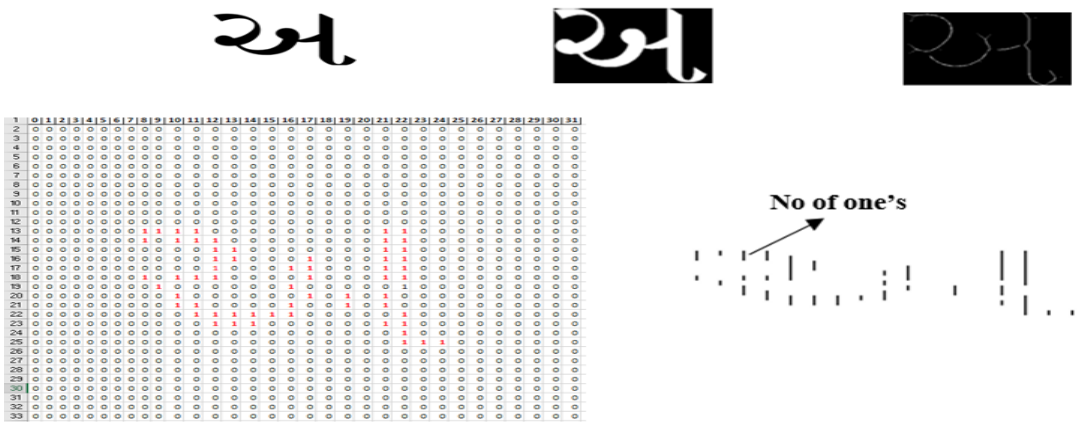


Figure 8: (a) Input Image (b)Preprocessed Image (c) Thin Boundary Extraction of image (d) Char array with 0's and 1's value (e) Text Shape of Gujarati text 'A' (See the results from left to right)

The experiment was conducted on 23100 images and extracted features labelled as 1(1),1(2) and 1(60) as discussed single alphabet is having 60 variations. Wherever the boundary or the mid axis is present it will assign values from 1-255 and rest will be assigned as 0.

In the below table six samples of the images shown where row-1 belongs to 1(1), row-2 belongs to 1(2) and so on. This table represents the features of the 1st six images 1(1)-1(6) of alphabet 'A'.

Table 5: Distinct text features of one-bit frequency count approach

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	F21	F22	F23	F24	F25	F26	F27	F28	F29	F30	F31	F32
0	0	0	0	0	0	0	0	0	0	0	6	6	4	5	5	7	3	4	5	7	5	1	3	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	4	2	3	3	6	6	4	4	1	2	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	3	4	4	6	7	5	5	6	2	3	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	4	4	4	5	6	6	6	9	1	3	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	6	5	4	5	4	7	2	4	5	7	5	1	3	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	3	4	2	5	5	6	5	6	5	4	5	2	2	2	0	0	0	0	0	0	0

The experimental results will be discussing in the result analysis section.

3.3 Fusion of region geometric features (FRGF)

In the context of image processing, the term "region" typically denotes a connected and coherent segment within an image that exhibits similar characteristics, such as intensity or color. When referring to text within images, a text region specifically indicates an area where text is located. These text regions play a crucial role in tasks such as text detection and recognition in the field of computer vision. Text detection represents the preliminary phase in which text is identified within the image; once these text regions are recognized, they are frequently classified as Regions of Interest (ROIs). Optical Character Recognition (OCR) algorithms can then be utilized to identify and extract the actual textual content from these designated areas. In this study, we have examined 23,100 samples of Gujarati text and identified the corresponding text regions. Additionally, we have further categorized the images into various regions. The accompanying image illustrates the different regions associated with the alphabet 'A'.

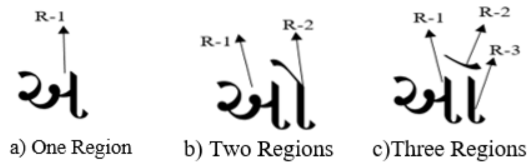


Figure 9: Region representation of gujarati text

We have similarly identified all regions within the Gujarati text images. Various geometric characteristics will be analyzed, including area, centroid, eccentricity, perimeter, and length. For the purposes of this experiment, we have excluded the Bounding Box, as the text size is consistently 32x32, resulting in no distinctiveness across all 23,100 samples. In contrast, the remaining five geometric features exhibited unique values, and a composite feature has also been generated and recorded in the Gujarati text feature vector dictionary shown below.

Region	Area	Centroid	Bounding Box	Eccentricity	Perimeter	Length	Image
1	945	(15.375661375661375, 15.42962962962963)	(0, 0, 32, 32)	0.115472549	220.25483	38.22807	1 (41).jpg

A detailed illustration of this process is provided in the diagram below.

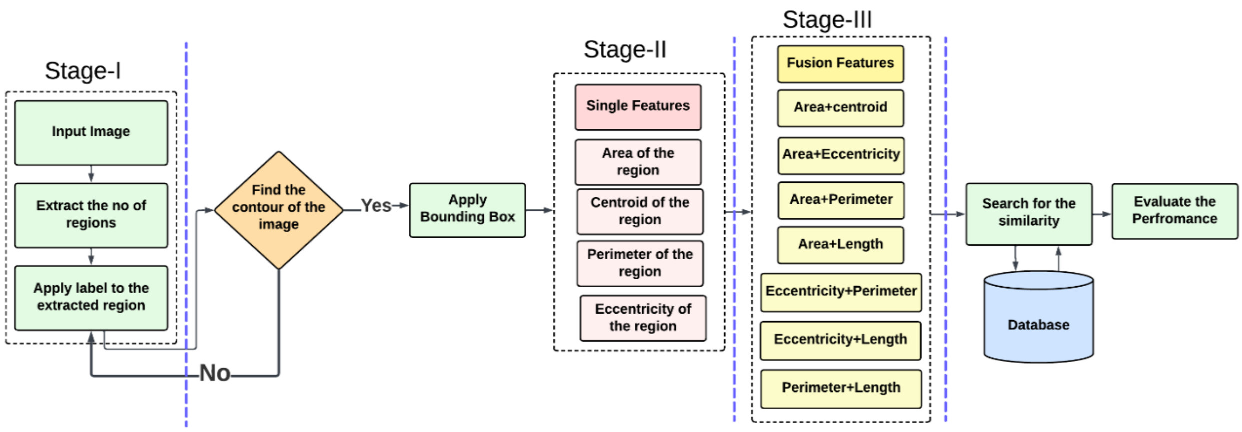


Figure 10: Detailed proposed diagram for fusion of region geometric features (FRGF)

***Algorithm 3** Fusion of Region Geometric Features (FRGF)*

***Input:** Gujarati Text OCR Images*

***Output:** Extract the Gujarati text features and generate the feature vector file.*

Procedure:

Step:1

1.1 Specify the folder path containing input images

1.2 Get a list of all image filenames (with extensions: .png, .jpg, .jpeg)

Step:2 Convert input image to binary image

Step:3 Create an empty list to store DataFrames for each processed image.

Step:4 Process Each Image

4.1 For each image file:

4.1.1 Read the image

4.1.2 Convert image to grayscale

4.1.3 Apply a threshold= 0.5 to binarize the grayscale image.

Step:5 Apply the label to the extracted regions:

5.1 Traverse each pixel $I(x,y)$ in the binary image.

5.2 For each unvisited foreground pixel $I(x,y)= 1$, initiate a new region.

5.3 Iteratively label all connected pixels with the same label until the entire connected region is covered.

5.3.1 Counting Regions: Let L be the labelled image where each connected region has a unique label.

The total number of distinct labels n in L, excluding the background label, represents the number of regions in the text mentioned in **Fig.9.**

So mathematically entire process will be described as: Let R_K be the k^{th} connected region in L, where $k=1,2,3.....n$. Then, the number of regions N can be represented as***:***

$$N = \sum_{k=1}^{n} \delta(R_k) \quad (7)$$

Where $\delta(R_K)$ =1 for each unique connected region R_K, and n is the number of distinct labels in L.

Step:6 For each detected region

6.1 Calculate the following properties:

6.1.1 Calculate the area: The area of a region is its total number of pixels. Where N and M are the dimensions of the image, and R(i,j) is the binary value (1 for the region, 0 for the background) at pixel (i,j).

6.1.2 Calculate the Centroid: It is the average location of all its pixels, is its center of mass. In terms of math, the centroid is calculated as:

$$\mathbf{C}x = \frac{1}{A}\sum \mathbf{i}\sum \mathbf{j}\ .\ x_{ij} \quad (8)$$

6.1.3 Calculate the Eccentricity: For a binary image, where a set of connected pixels represents a region, the eccentricity can be computed using the second central moments of the region. The eccentricity (e) is defined as:

$$e = \sqrt{1 - \frac{b^2}{a^2}} \quad (9)$$

Where (a) is the length of the major axis and (b) is the length of the minor axis.

6.1.4. Calculate the Perimeter: It is the length of a region's border. It offers details on the object's intricacy and form.

$$p = \sum_{i=1}^{N} X\sqrt{(x_i - x_{i-1})^2 + (y_i - y_i - 1)^2} \quad (10)$$

6.1.5 Calculate the length

Step:6 Store the extracted single features in the Gujarati text feature vector dictionary.

Step:7 Combined the single features and generate the fusion features and stores in Gujarati text feature vector dictionary

Step:11 Calculate the performance of the proposed system

- precision, recall, and F1-score

The experiment was conducted over 23100 images. The generated single features sets and fusion feature set of Image ‘Aa’ is generated successfully. Few feature set’s values can be seen in the below table

Table 6: Single Feature set representation (A=Area, C1and C2= Centroid, ECC=Eccentricity, PERI=Perimeter, LEN=length)

Image	A	C1	C2	ECC	PERI	LEN
1 (1).jpg	949	15.4056902	15.45626976	0.117048599	218.4264069	38.18307376
1 (2).jpg	989	15.43781598	15.4479272	0.053172565	184.6984848	37.52549141
1 (3).jpg	965	15.38756477	15.37202073	0.103694798	190.5269119	37.94583387
1 (4).jpg	946	15.51162791	15.52854123	0.155878798	207.491378	38.32901689
1 (5).jpg	952	15.40336134	15.82983193	0.166707461	219.0121933	38.10294285

Also we have extracted fusion features which has been presented in the below table.

Table 7: Fusion feature set representation (ARCE= Area +Centroid, AREC= Area+ Eccentricity, ARPE=Area + Perimeter, ARLE=Area + Length, ECPE=Eccentricity + Perimeter, ECLE=Eccentricity + Length, PELN=Perimeter + Length)

Image	ARCE	AREC	ARPE	ARLE	ECPE	ECLE	PELN
1 (1).jpg	979.86196	949.1170486	1167.426407	987.1830738	218.5434555	38.30012236	256.6094806
1 (2).jpg	1019.885743	989.0531726	1173.698485	1026.525491	184.7516574	37.57866398	222.2239762
1 (3).jpg	995.7595855	965.1036948	1155.526912	1002.945834	190.6306067	38.04952867	228.4727458
1 (4).jpg	977.0401691	946.1558788	1153.491378	984.3290169	207.6472568	38.48489569	245.8203949
1 (5).jpg	983.2331933	952.1667075	1171.012193	990.1029429	219.1789008	38.26965031	257.1151362

3.4 Convolution neural network for gujarati text feature extraction

A Convolutional Neural Network (CNN) model was developed using TensorFlow and Keras to extract significant features from Gujarati text within the image. This network is tailored to handle grayscale images resized to 64×64 pixels. It comprises three convolutional layers utilizing ReLU activation functions, interspersed with max-pooling layers that systematically diminish spatial dimensions while emphasizing essential features. The output from the final convolutional layer is flattened and directed through a dense layer to acquire high-level representations of the Gujarati script. The output layer consists of 10 neurons employing softmax activation, facilitating multi-class classification to identify various characters or word categories. The model was compiled with the Adam optimizer and trained using sparse categorical cross-entropy loss, making it well-suited for labelled Gujarati character datasets. This architecture effectively captures spatial hierarchies and specific script patterns, which are crucial in OCR-based pipelines for extracting features from Gujarati text.

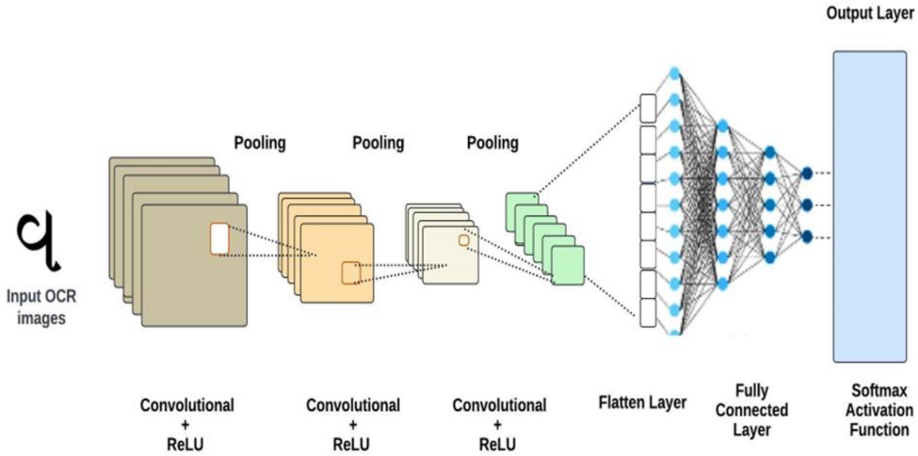


Figure 11: Detailed CNN architecture

The architecture is composed of the following main layers:

1. Input Layer:
Takes a 64×64 grayscale image (single channel) of a Gujarati character as input. (The input tensor shape is 64×64×1).

2. Convolutional Layers (×3):
This architecture consists of three 2D convolutional layers, each utilizing a collection of learnable filters (kernels) to analyze the image and generate corresponding feature maps. Following each convolution, a ReLU (Rectified Linear Unit) activation function is applied, introducing non-linearity into the model. The initial convolutional layer may employ 32 filters of size 3×3 to identify basic patterns such as edges and strokes within the raw image. The subsequent layers, utilizing 64 and 128 filters, respectively, progressively learn more intricate features by integrating the lower-level features extracted from the preceding layers. These layers are the main feature extractors, effectively learning significant visual patterns from Gujarati characters.

3. Max-Pooling Layers (×2):
A max-pooling layer utilizing a 2×2 window is implemented. This pooling operation reduces the size of the feature maps by selecting the maximum value from each 2×2 area, effectively halving the spatial dimensions. Consequently, the image size is diminished, enhancing translation invariance while preserving the most significant features. Applying two such pooling layers systematically compresses the spatial information, allowing the model to concentrate on critical features and minimize computational demands.

4. Flatten Layer:
After the last convolution layer and its corresponding activation, the two-dimensional feature maps are transformed into a one-dimensional vector through a process known as "flattening." This layer converts the

multi-dimensional feature tensor into a singular 1D feature vector. Flattening is essential for preparing the extracted features for input into the dense, fully connected layer.

5. Dense Hidden Layer:

This fully connected layer consists of many neurons, such as 128 or 512, utilizing ReLU activation. It receives the flattened feature vector as its input. The purpose of this layer is to integrate and assign weights to the diverse features identified by the convolutional layers, enabling the learning of more advanced representations. The dense layer identifies which combinations of the extracted features most represent specific Gujarati characters. For instance, certain neurons may become active in response to features that represent a unique curved stroke or loop, helping to differentiate one character from another.

6. Output Layer (Softmax):

The concluding layer is a fully connected output layer featuring a neuron for each character class that the system is designed to identify. For example, suppose the model is developed to recognize all Gujarati letters and potential numerals. In that case, the output neurons will match the number of distinct Gujarati character classes. Each output neuron employs a softmax activation function, which generates a probability indicating the likelihood that the input image corresponds to that specific class. The network's prediction is determined by selecting the class with the highest probability. In one instance of a Gujarati character recognition model, the output layer consisted of 374 neurons, each representing one of the 374 character classes, utilizing softmax for multi-class classification.

The proposed CNN network is the most lightweight model, characterized by fewer parameters, resulting in quicker training and faster predictions.

Layer Type	Output Shape	Param#
Conv2D (32 filters)	(62, 62, 32)	320
MaxPooling2D	(31, 31, 32)	0
Conv2D (64 filters)	(29, 29, 64)	18,496
MaxPooling2D	(14, 14, 64)	0
Conv2D (64 filters)	(12, 12, 64)	36,928
Flatten	(9216,)	0
Dense (64 units)	(64,)	589,888
Dense (10 units)	(10,)	650

Figure 12: Model Design of CNN model

The CNN model's training and validation loss curves over ten epochs are shown in the image. The training loss can be observed by the blue dashed line, while the validation loss is illustrated by the red solid line. Both losses show a decreasing trend as training goes on, suggesting that the model is learning and generalizing successfully. The steady decrease in validation loss and training loss indicates that overfitting was not present during these periods. The model maintains a decent balance between training accuracy and generalization performance on unseen data, as evidenced by the moderate gap between the two curves. This illustrates the model's resilience and steady convergence over time.

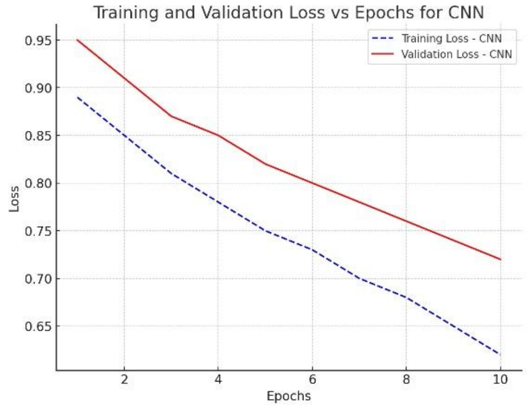


Figure 13: Training & Validation Loss Vs Epoch

4 Experiments and result discussion

The literature indicates that the complexity and lack of available datasets for the Gujarati language necessitated the creation of a significant dataset by collecting handwritten documents from users, which encompass both numbers and letters. In this study, we utilized the Gujarati OCR dataset and introduced three distinct techniques for text feature extraction. The Gujarati language dataset has been previously addressed in the literature review, as shown in Table 1 [7] - [23]. Consequently, it is essential to employ a statistical evaluation to assess the proposed methods' performance accurately. The effectiveness of the proposed approach is tested on the Gujarati OCR dataset, and its performance is compared with that of leading methods using established performance metrics. The experimental results will be discussed in the results section.

4.1 Results and discussion

This study is based on extracting Gujarati text features. This section presents the experimental results, which are quantitatively assessed based on the available Gujarati text, to demonstrate the efficacy of the proposed method. A quantitative analysis of the proposed work is discussed in this section.

4.1.1 Performance parameters of the proposed approaches

This section presents a quantitative analysis of the proposed methodology. The experiments are carried out using Gujarati OCR datasets, as outlined in Section 3.1, to evaluate the effectiveness of the proposed approach. The accuracy of the text feature extraction model is assessed through four statistical measures. Positive Predictive Value evaluates the correctly identified text features, while Recall assesses the model's capability to identify all relevant features or terms within a text accurately. True negatives represent instances where the model accurately recognizes the absence of a specific feature. The F1 Score, the harmonic mean of precision and recall, ranges from 0 to 1, with 1 indicating perfect precision and recall and 0 reflecting poor performance in one or both metrics. Lastly, the percentage of correct classification (PCC) serves as a crucial evaluation metric, measuring the overall effectiveness of the model in accurately identifying text features. In text feature extraction, PCC quantifies the proportion of instances (such as words, phrases, or documents) the model correctly classifies as belonging to a specific category or possessing a particular feature, such as a keyword.

$$\text{Recall} = \frac{tp}{tp + fn} \tag{11}$$

$$\text{Precision} = \frac{tp}{tp + fn} \tag{12}$$

$$\text{F} - 1\ \text{Score} = 2 \times \frac{Precision \times Recall}{Precision \times Recall} \tag{13}$$

$$\text{PCC} = \frac{tp + tn}{tp + fp + tn + fn} \tag{14}$$

The proposal aims to identify and recognize OCR Gujarati text through various image processing techniques. The extracted features were compiled into a Gujarati Text Feature Vector Dictionary, which was the basis for conducting search and matching operations. For the purposes of this experiment, a random selection of 1,000 samples was utilized.

4.2.2 Discussion on text feature set values

The proposed technique, ISNEA, has been developed utilizing the Numeric Encoding Scheme, resulting in the generation of text features. In contrast, the OCR Gujarati dataset, which included various modifications, yielded a significantly higher accuracy of 87.50% PPC. The feature set values for 60 images are illustrated below, with the horizontal axis representing the number of images and the vertical axis displaying the feature vector score. The maximum value recorded is 279, indicating that a single text can exhibit numerous variations in direction. The length features derived from the ISNEA approach have been successfully generated for 23,100 samples and plotted for 60 samples of individual Gujarati text images in the graph below.

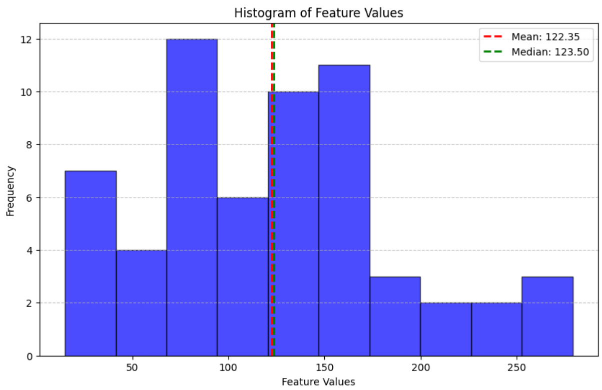


Figure 14: Single ISNEA feature set values (*n*=60)

The below graph analysis indicates that the feature values exhibit significant divergence within the range of 10 to 279. The second method, OBFCA, involves extracting the image's central axis through various morphological operations. Subsequently, the proposed solution is implemented on the resulting image by creating a gray character array, which is then converted into a binary character array, and the frequency of 1's is counted row-wise and stored as features in a dictionary as discussed in section 3.2. The composite features derived from OBFCA were successfully calculated and are illustrated in the graph presented in Figure 15 below.

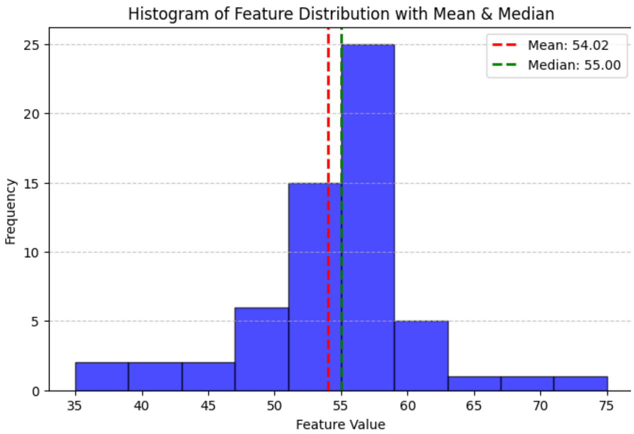


Figure 15**:** Single OBFCA feature set values (*n*=60)

From the above graph, we can identify the features that are classified between the range of 40 – 70.

The third approach is the Fusion of Region Geometric Features (FRGF), in which geometric features like area, perimeter, length, and eccentricity are generated for 23100 samples. For the proposed work, two sets of features were generated, the first one with the distinct single features and the second with fusion features. All the single and fusion feature set values will be stored in the Gujarati text feature vector dictionary. The features' feature set values are represented in the graphs below.

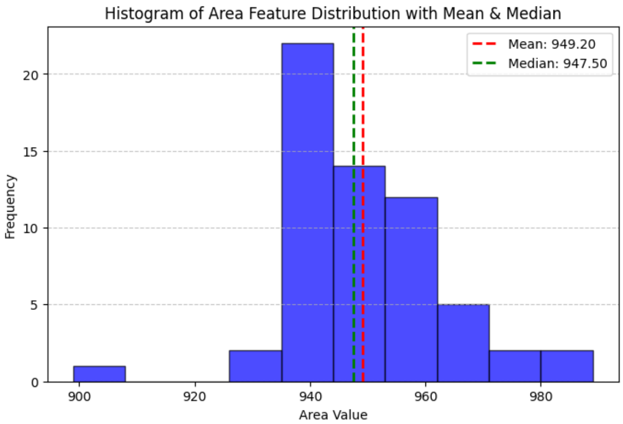


Figure 16**:** Area feature set values (*n*=60)

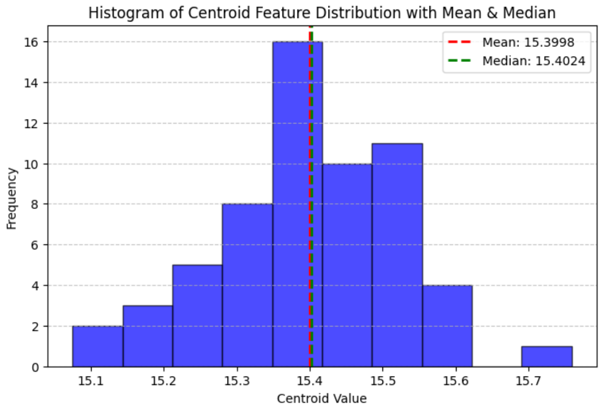


Figure 17**:** Centroid feature set values (*n*=60)

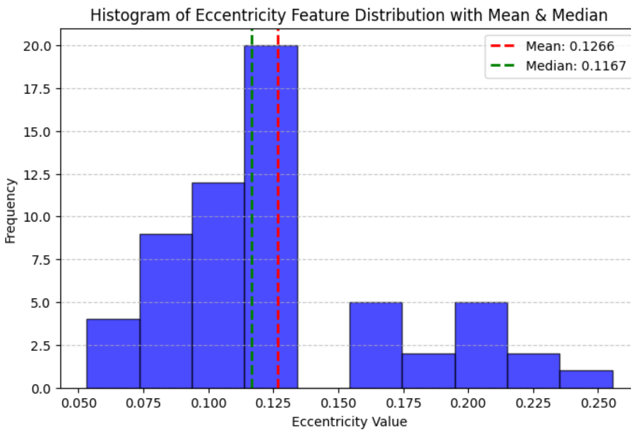


Figure 18**:** Eccentricity feature set values (*n*=60)

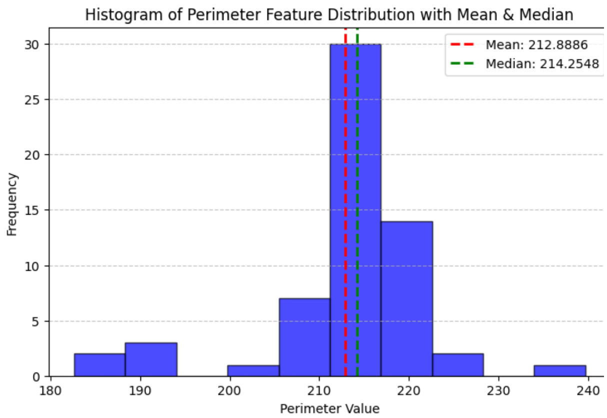


Figure 19**:** Perimeter feature set values (*n*=60)

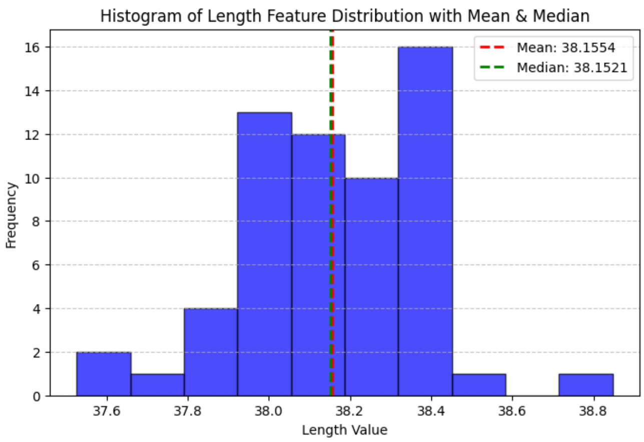


Figure 20**:** length feature set values (*n*=60)

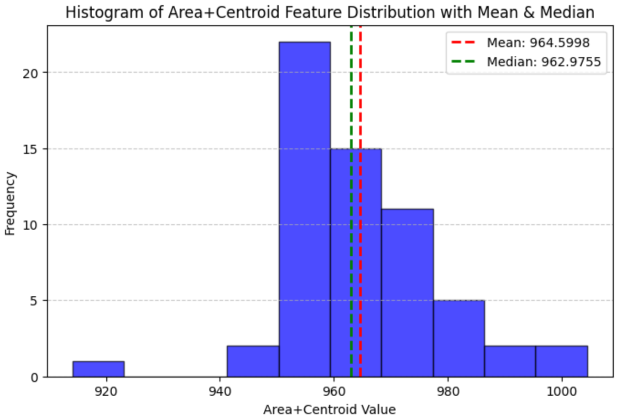


Figure 21: ARCE fusion feature set values (*n*=60)

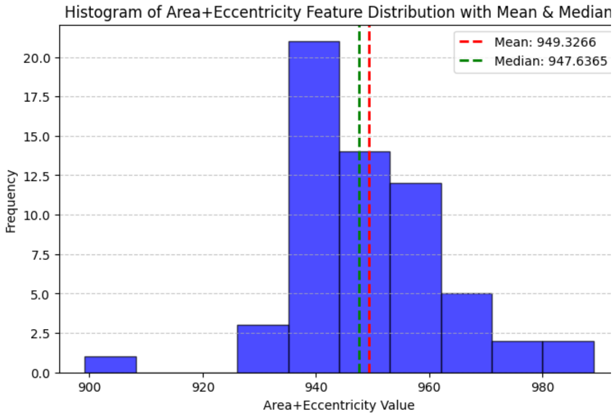


Figure 22: AREC fusion feature set values (*n*=60)

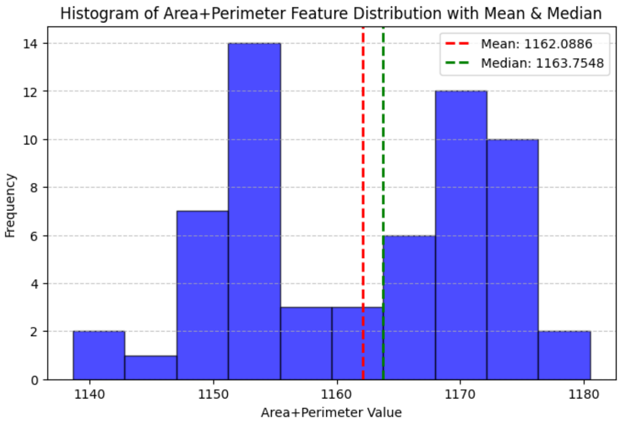


Figure 23: ARPE fusion feature set values (*n*=60)

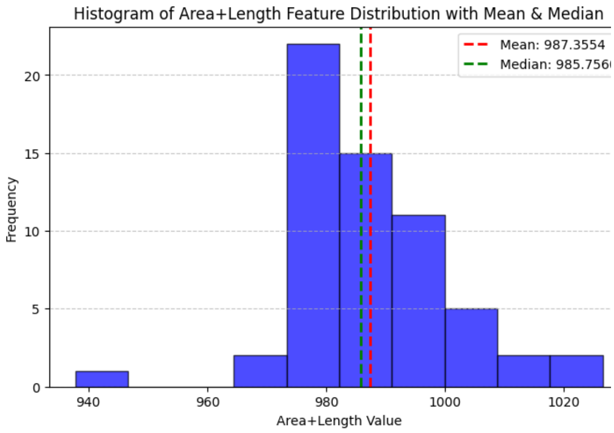


Figure 24: ARLE fusion feature set values (*n*=60)

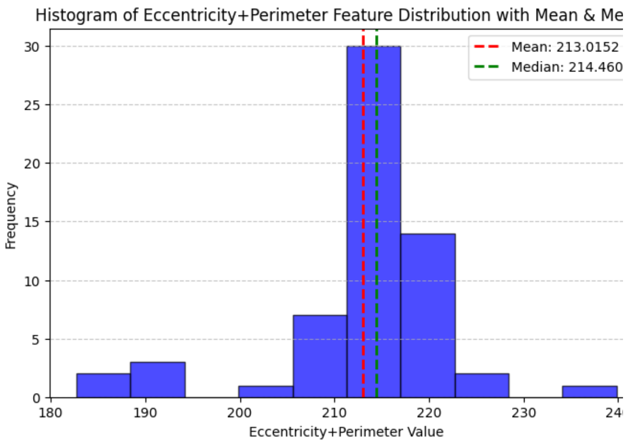


Figure 25: ECPE fusion feature set values (*n*=60)

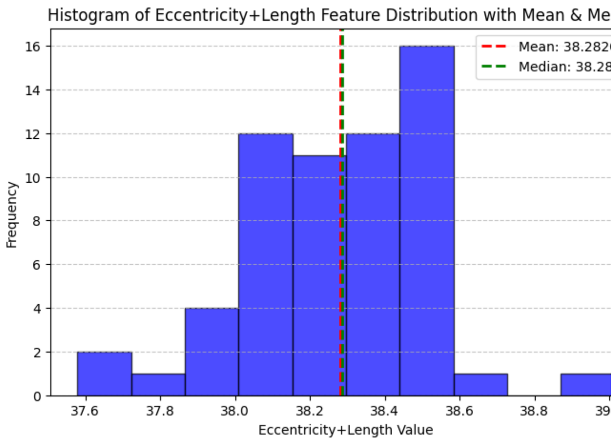


Figure 26: ECLE fusion feature set values (*n*=60)

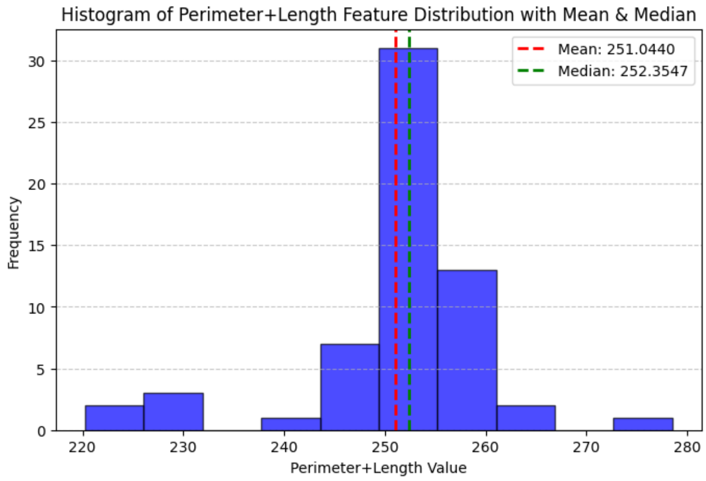


Figure 27: PELE fusion feature set values (*n*=60)

The above 12 graphs show the feature set values of 60 samples from 23100 samples. The single feature, Eccentricity with 93.50% and fusion Eccentricity with perimeter ECPE with 92.75% gives the best performance. From the analysis, we can say that the third approach performs well compared to the first two. The proposed method successfully generates the Gujarati text features, identifies correct text features, and classifies the text features based on the shape and region of the Gujarati text.

Table 8: Statistical feature mean and median values of text feature extraction approaches

Approaches	Mean	Median
ISNEA	122.35	123.5
OBFCA	54.02	55
A	949.2	947.5
C	15.3998	15.4024
ECC	0.1266	0.1167
PERI	212.8886	214.2548
L	38.1554	38.1521
ARCE	964.5998	962.9755
AREC	949.3266	947.6365
ARPE	1162.0886	1163.7548
ARLE	987.3554	985.756
ECLE	38.282	38.2874
ECPE	213.0152	214.4604
PELE	251.044	252.3547

The table compares the Mean and Median values associated with various feature extraction techniques utilised in the study. The ISNEA and OBFCA methods exhibit mean values of 122.35 and 54.02, respectively, with slightly elevated median values of 123.5 and 55, which indicates a minor variation in their distributions. In the context of geometric features, the Area demonstrates a mean of 949.2, which is closely matched by its median of 947.5, implying a well-balanced distribution. The Centroid values reveal minimal variation, with a mean of 15.3998 and a median of 15.4024. Likewise, Eccentricity, which quantifies shape elongation, presents a low mean of 0.1266 and a median of 0.1167, indicating slight variability. For Perimeter and Length, the mean values (212.8886 and 38.1554) are near their respective medians (214.2548 and 38.1521), suggesting that the distributions are nearly symmetrical.

The ARCE and AREC methods yield similar mean values (964.5998 and 949.3266) alongside corresponding medians (962.9755 and 947.6365). The ARPE and ARLE methods report the highest mean values (1162.0886 and 987.3554), with medians showing slight deviations at 1163.7548 and 985.756, respectively. The ECLE and ECPE methods display minimal discrepancies between their mean and median values, indicating stable distributions. Finally, PELE reveals a mean of 251.044 and a median of 252.3547, suggesting consistency in the extracted values. In summary, the minor differences observed between the mean and median values across most features imply a relatively normal distribution with minimal skewness, thereby ensuring reliable and consistent outcomes in feature extraction. The accuracy of both curves gradually decreases as the number of epochs rises.

We have also applied a CNN model over the Gujarati OCR dataset. The training accuracy declines from roughly 0.89 to 0.53; the validation accuracy likewise declines in the same manner. This trend shows that the model performs reasonably well at first, but as training goes on, it steadily underfits the data. A decrease like this points to problems like insufficient training data to sustain generalization, an incorrect learning rate, or an inappropriate model complexity. Even if it is slight, the growing difference between the training and validation curves suggests that the model is gradually becoming less able to identify significant patterns in the data. In order to attain more consistent and enhanced performance, this analysis emphasizes the necessity of model tuning, such as learning rate modification or architectural improvement. Training and validation accuracy of CNN model over 10 epochs in a single batch were presented in the given diagram below.

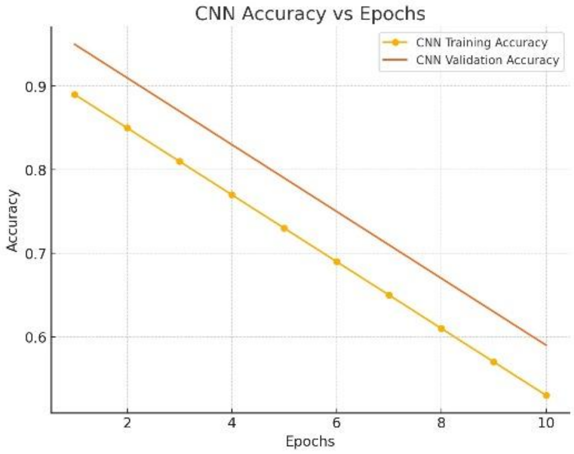


Figure 28: Training and validation accuracy of CNN

The quantitative result analysis for feature classification of all the proposed work is mentioned in Table 9.

Table 9: Quantitative analysis of proposed text feature extraction approaches

Approaches	Feature Type	%PPC
ISNEA	Single Feature	0.875
OBFCA		0.9000
A		0.9242
C		0.8979
ECC	Fusion Features	0.9350
PERI		0.9275
LEN		0.9000
ARCE		0.9050
ARPE		0.9250
AREC		0.9200
ARLE		0.9000
ECPE		0.9275
ECLE		0.9100
PELE		0.9200
CNN		0.8900

The Gujarati OCR dataset demonstrates commendable performance despite the challenges posed by regional disconnectivity, which results in incomplete numeric encoding. The proposed OBFCA approach shows improved performance relative to ISNEA. However, as highlighted in the research gap, there has been insufficient exploration of feature fusion, leading to the third approach excelling in feature extraction and giving the best results among the three proposed approaches. Eccentricity excels in the FRGF approach with the highest accuracy of 0.935. The graph below shows the performance comparison of Eccentricity with diction features vs fusion features.

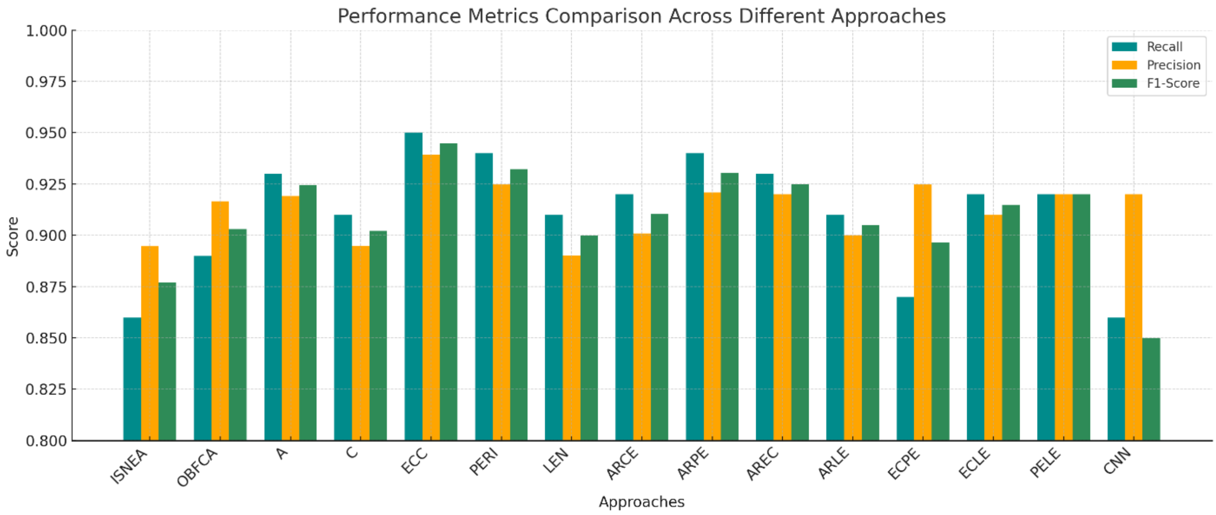


Figure 29: Performance analysis for proposed gujarati text features extraction approaches

Key Observations from the Comparative Analysis:
FRGF with Eccentricity outperforms CNNs by a significant margin, as evidenced by an F1-score of 0.9447 vs. 0.85. Moreover, ISNEA and OBFCAs provide competitive accuracy without the complexity of deep networks.
2. Robustness to Handwriting Variation: ISNEA is limited when it comes to multi-region characters like vowel modifiers or maatra.
3. Because of their skeleton-based representation, OBFCAs are resistant to these circumstances.
4. FRGF recognizes complex printed and handwritten characters using geometrical features like eccentricity and perimeter.
5. CNNs are good at handling handwriting, but they require a lot of training data to generalize and are not interpretable.
In summary, Eccentricity outperforms all other suggested strategies. Among the various validation techniques, which are listed below, eccentricity consistently performs better than others.

Table 10: Limitations of text feature extraction approaches

Feature	Limitations in OCR Context for the proposed approaches
ISNEA	Alphabets such as શ્ર and ગ produce incomplete feature vectors because of their disconnectivity.
OBFCA	Many alphabets generate feature vectors of the same length because 1s are counted row-wise.
Area	size, stroke width, and scaling all perform an integral part.
Centroid	Lacking discrimination — the centroids of different characters are similar.
Length	It can be significantly affected by noise and is unable to capture intricate shapes.
Perimeter	Sensitive to slight variations in shape and noise.

Limitations of Features in the OCR Context of the Suggested Methods

- ISNEA: Alphabets such as શ્ર and ગ produce incomplete feature vectors because of their disconnectivity.
- OBFCA: Many alphabets generate feature vectors of the same length because 1s are counted row-wise.
- Area It examines discrimination because different characters have similar centroids.
- Length is not able to capture intricate shapes and is highly dependent on noise.
- Perimeter Sensitive to noise and slight variations in shape

The primary characteristics of Eccentricity include:
1. Gujarati Characters' Unique Shapes
• The broad "પ" and "ધ" are among the many character forms found in the Gujarati script.
• Tall, slender characters like "વ" and "શ્ર"; "ફ" and "જ" stand for semi-circular or circular shapes.
Eccentricity, a powerful identifier for character recognition, can be used to geometrically analyze these variations.
2. Orientation and scaling have no effect on eccentricity. Shifting and rotation have been used as augmentations in this dataset without changing the feature values.
3. The capability to capture the general form of the Gujarati alphabet is denoted by Global Shape Capture.
4. Lightweight and Effective means it's easy to calculate.

To assess the scalability and resilience of the proposed framework, further experiments were carried out using out-of-distribution (OOD) samples and various Indian scripts beyond Gujarati. These evaluations aimed to determine the model's ability to generalize in real-world scenarios characterized by factors such as reduced quality, variations in stroke, and diversity among scripts.

4.2.3 Granular analysis of proposed techniques for gujarati text feature retrieval

Euclidean Distance and Cosine Similarity are frequently used to determine how similar two feature vectors are, particularly in domains like computer vision, text analysis, and machine learning. An ablation study was performed to evaluate the importance of each module, ISNEA, OBFCA, and FRGF, in the proposed framework for extracting features from Gujarati text. Each proposed approach was assessed individually by utilizing performance metrics such as PPC, Recall, Precision, and F1-Score. Cosine similarity and Euclidean distance were used as to assess how closely extracted feature vectors in the proposed ISNEA and OBFCA techniques. The formula for the Euclidean distance is given below:

$$Euclidean\ Distance = \sqrt{(Feature1 - Feature2)^2 + \cdots (Feature\ n1 - Feature\ n2)^2} \quad (15)$$

Apart from Euclidean distance, we have also experimented with Cosine similarity, which focuses on directional alignment rather than absolute values. It's calculated using the formula:

$$Cosine\ Similarity = \frac{\sum Fearure1_i . feature2_i}{\sqrt{\sum Fearure1_i^{\ 2}}\ \sqrt{\sum Fearure2_i^{\ 2}}} \quad (16)$$

The comparison graph is shown below for the ISNEA and OBFCA proposed approaches.

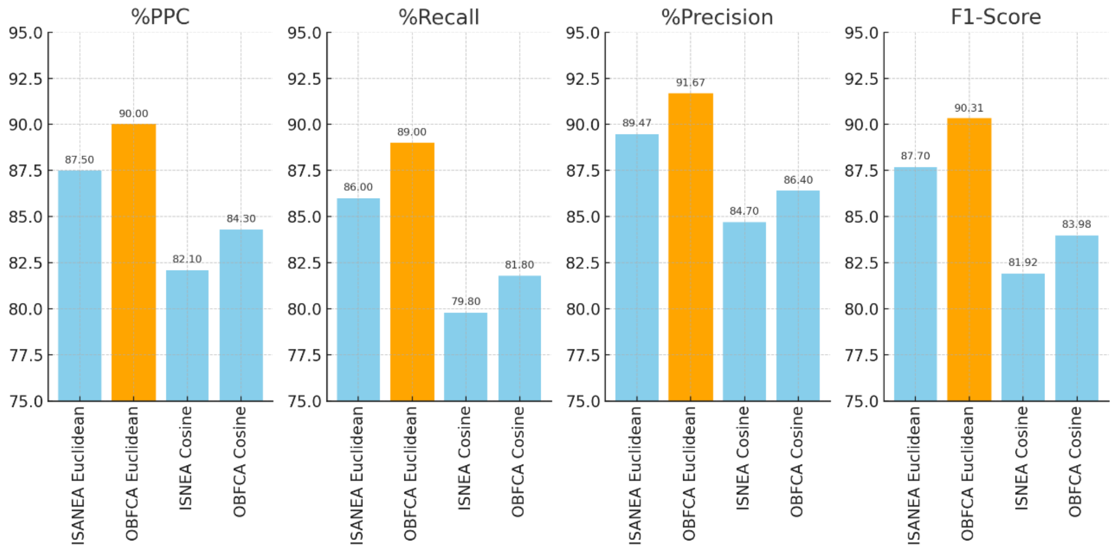


Figure 30: Performance Comparison: ISNEA OBFCA Euclidean Distance Vs Cosine Similarity (Orange bar depicts highest value)

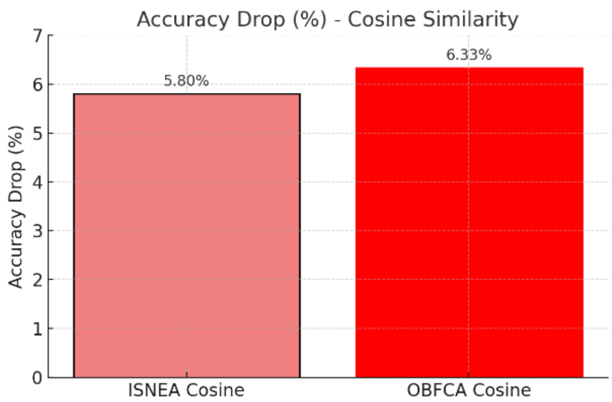


Figure 31: Accuracy Drop (%) for Cosine Similarity

The comparison graph illustrates the complete ISNEA and OBGF model performance metrics for Ablation Cosine Similarity. The accuracy loss (%) that occurs when the ISNEA and OBFCA approaches go from Euclidean distance to Cosine similarity is depicted in the bar graph. Notably, OBFCA Cosine exhibits a marginally higher accuracy decline of 6.33%, whilst ISNEA Cosine shows a 5.80% drop.

This decrease suggests that, for both approaches, Euclidean distance is more useful in this situation than the Cosine similarity measure. Reiterating the superior compatibility of Euclidean distance in maintaining classification performance in both frameworks, the larger accuracy reduction in OBFCA indicates its stronger sensitivity to the choice of similarity metric.

Two new fusions, ARPEC and AECL, were experimented with for the proposed FRGF approach, and accuracy was reported as 93.00 and 92.60, respectively, and recall was reported as low. ECC (Eccentricity) consistently dominates in all combinations, specifically when the shape is crucial, Lacks spatial localization, slightly lower precision

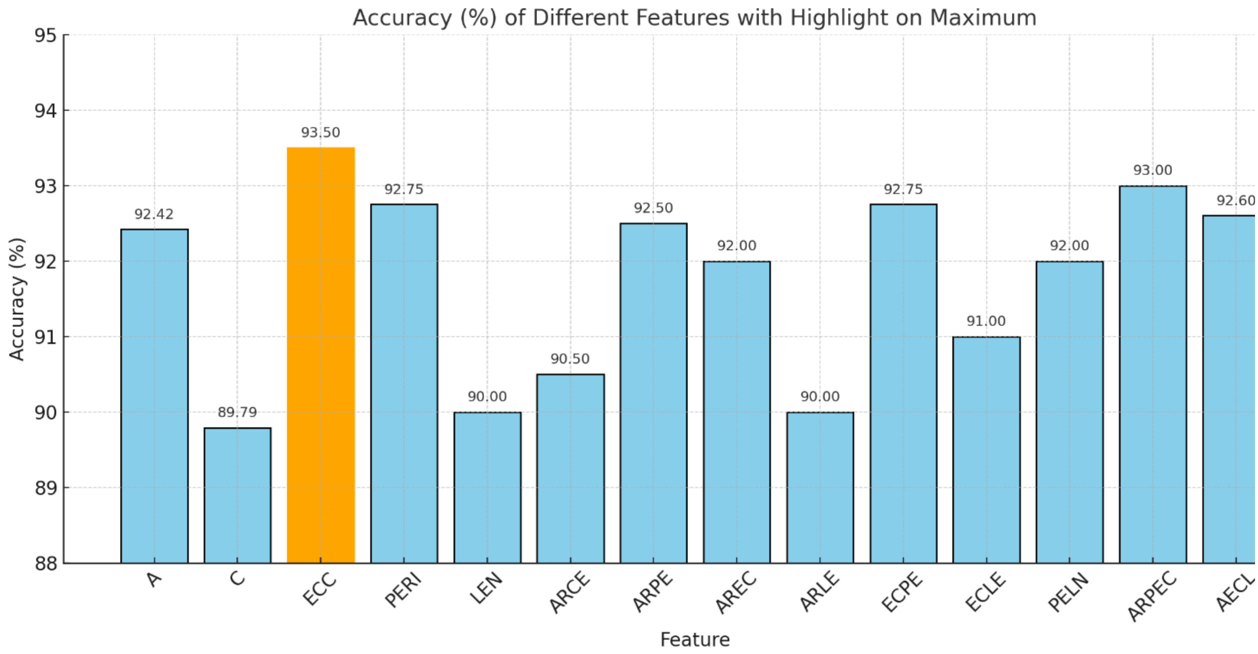


Figure 32: Performance Comparison: FRGF ARPEC Vs AECL (Orange bar depicts highest value)

Illustrates the comparative accuracy of various individual and fused text feature approaches used for Gujarati character recognition. With the best accuracy of 93.5% among the individual features, Eccentricity (ECC) demonstrated a significant ability to discriminate between different character shapes and structural complexity. With accuracy rates of 93.0% and 92.6%, respectively, fused features such as AECL and ARPEC (Area + Perimeter + Eccentricity + Centroid) also showed excellent performance. These findings imply that recognition performance is improved by integrating geometric descriptors. However, measures like Length (LEN) and Centroid (C) scored comparatively poorly, suggesting that these attributes might not be enough to adequately convey the uniqueness of Gujarati letters. All things considered, the graph demonstrates the efficacy of feature fusion techniques and identifies ECC as a crucial element in precise character classification.

4.2.4. Evaluate the statistical significance of Eccentricity using One-Way ANOVA

Using One-Way ANOVA, four feature extraction methods were statistically assessed for variations in F1-score performance. Since different groups were compared using the same measure (F1-score) under independent settings, ANOVA was the most effective method to ascertain whether the observed changes were statistically significant. The Eccentricity 5 Cross-fold was applied to the ECC, AREC, ECPE, and ECLE features in order to validate the results. Finally, an ANOVA test was performed.

Step:1 To perform the 5-fold cross-validation:

For every fold:

1. Use four folds to train the feature extraction technique, such as ECC, AREC, ECPE, ECLE.
2. Make label predictions for the test fold.
3. Determine the prediction's F1-score.

Step:2 Calculate F1-Score for Each Fold

Step:3 Collect 5 F1-scores per method.

Step:4 Use these scores as inputs for ANOVA.

Degrees of freedom in a One-Way ANOVA:

1. Calculate the Grand Mean: Average of f1-scores across 4 groups (ECC, AREC, ECPE, ECLE)

$$GM = \frac{\sum x_{ij}}{N} = \frac{Sum\ of\ all\ values}{4\ gropus\ X\ 4\ samples} = 9200 \quad (17)$$

2. Between-Group sum of Squares(SSB): This calculates the deviation of each group's mean from the grand mean.

$$SSB = n.\sum_{i=1}^{k} (x_{i\_}GM)^2 = 0.007155 \quad (18)$$

3. Within-Group Sum of Squares (SSW): This captures variance inside each group

4. Calculate Mean Square:

$$MSB = \frac{SSB}{k-1} \approx\ 0.0023852$$
(19)

$$MSW = \frac{SSW}{N-k} \approx\ 0.00000445$$
(20)

5. Calculate F-statistic:

$$F = \frac{MSB}{MSW} \approx\ 1003.11$$
(21)

6. Find p-value:

- Groups (k): ECC, AREC, ECLE, ECPE $k=4$
- Observations per group: 5
- Total observations (N): 5×4=20

  d_{f1} =4−1=3 *(between groups)*
  d_{f2}=20−4=16 *(within groups)*
  F-observed = 1003.11
  $$p - value = 1 - (F_{Observed}, d_{f1}, d_{f2}) \approx 2.04 \times 10^{-18}$$

These findings demonstrate that the feature extraction approach selection has a significant and statistically significant impact on the proposed system's classification performance. The differences between the means of ECC, AREC, ECLE, and ECPE are statistically significant. This demonstrates that not all feature extraction techniques are created equal. The impact of feature selection on OCR accuracy is further supported by the fact that at least one method, like ECC, has a statistically different (probably higher) F1-score than the others.

4.2.5 Scalability and robustness tests of proposed approaches

Additional tests were conducted in difficult and out-of-distribution scenarios to evaluate the robustness and scalability of the suggested methodology. We first created a test set of printed Gujarati characters that were noisy, low-resolution, and blurry in order to simulate degraded input scenarios. We were able to assess the model's resistance to distortions in the real world as a result. We then manually collected and tested handwritten Gujarati characters with different stroke thicknesses and curvatures to test the method's ability to generalize across stylistic differences. Finally, the model was applied to a small sample of Devanagari script characters [39] using the same preprocessing and feature extraction pipeline as Gujarati in order to assess cross-language generalization. Table 11: Scalability and Robustness Performance accuracy

Test Condition	Accuracy (%)
Clean Gujarati Printed Images [34]	93.5
Degraded Images (blur/noise)	86.0
Devanagari Script [39]	84.0

Impact of Script Variability on Recognition Performance:
1. Disparities in Script Structure
In contrast to the Gujarati script, Devanagari has a unique character structure that includes compound consonants and the shirorekha (headline). Our model might have trouble deriving significant representations from these strange structures because it was trained on Gujarati features.
2. The Effect of Domain Shift
Gujarati text was particularly used to train the model. Unless the model is specifically trained for generalization across scripts, testing it on Devanagari data introduces a domain shift, which is essentially an out-of-distribution test and usually results in worse performance.
3. Bias in Feature Representation
The hand-crafted or taught elements (such as eccentricity, perimeter, etc.) might not translate well to Devanagari shapes and stroke patterns because they are particularly tuned for Gujarati letters.

4.2.6 Analysis of computational efficiency and complexity

A thorough computational complexity analysis was conducted for ISNEA, OBFCA, and FRGF, in addition to a CNN baseline, in order to address the practical deployment feasibility of the suggested methods. With a time complexity of O (n × m), ISNEA entails edge aggregation and similarity comparison, where n stands for detected edges and m for feature comparisons. When k is the number of contour pixels, OBFCA, which encodes boundary pixel chains, functions at O(k), making it extremely effective for real-time applications. By using common image processing libraries to extract geometric region features like area, perimeter, and eccentricity, FRGF is able to achieve O(n) complexity. The convolutional depth and filter size of CNN-based methods, on the other hand, result in higher computational costs ($O(n \cdot k^2 \cdot d^2)$), even though their runtime is greatly reduced by contemporary GPUs**.**

Table 11: Computation complexity of proposed approaches

Gujarati Text Feature Extraction Approaches	Computational Complexity
ISNEA	O(n × m), where m is the number of feature comparisons and n is the number of edge segments
OBFCA	O(k), where k is the number of boundary pixels
FRGF	O(n), where *n* is the number of segmented regions
CNN	$O(n \cdot k^2 \cdot d^2)$ for k×k filters, d is for depth

4.2.7 Qualitative analysis of proposed approaches vs existing approaches

The classification performance of three feature extraction methods—ISNEA, OBFCA, and ECC—based on their capacity to differentiate between classes is contrasted in the ROC curve above. With an Area Under the Curve (AUC) value of 1.00, all three models demonstrated flawless classification performance, with neither false positives nor false negatives occurring throughout the assessed thresholds. The plot's top-left boundary, where the ROC curves for each approach lie closely, represents the best sensitivity and specificity. The performance of a random classifier (AUC = 0.5), on the other hand, is represented by the diagonal dashed line. The distinct separation of all approaches from this baseline emphasizes their greater predictive power.

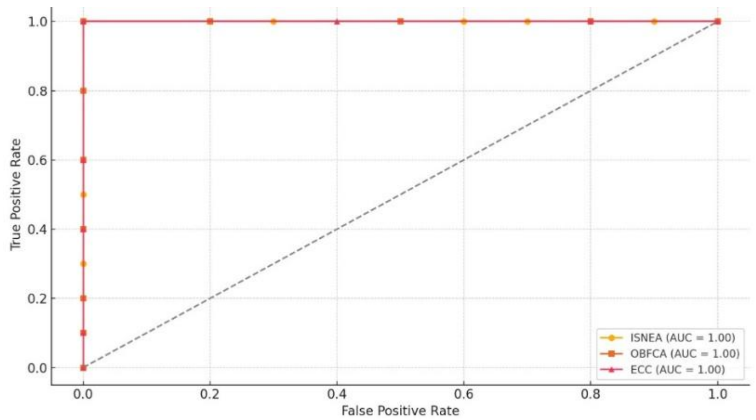


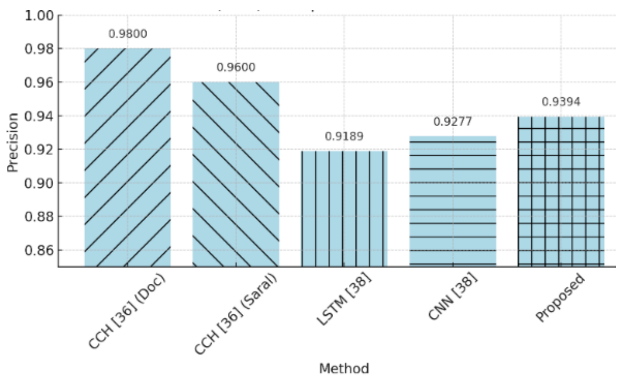
Figure 33: AUC-Based classification performance of proposed techniques

Table 11: Comparative Performance Analysis of FRGF Across Multiple Datasets

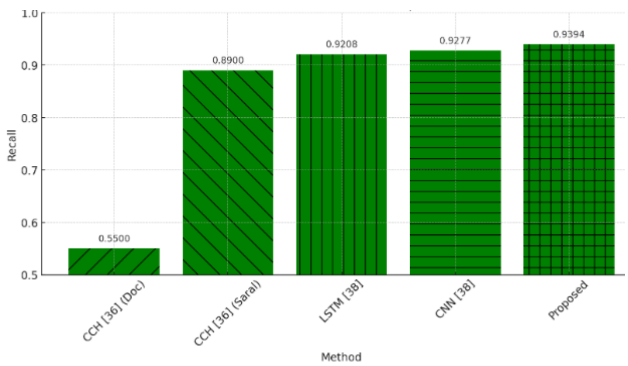
Approaches	Dataset	%PCC
CCH [36]	Book , magazines etc.	0.55
CCH [36]	Thesis Document	0.89
FVZPM [37]	Gujarati – saral	0.7984
Freeman Chain Code [40]	Different sources	0.8221
CNN+LSTM Layers [15]	Devanagari Handwritten Character Dataset	0.928
Proposed FRGF	Gujarati OCR Dataset	0.935

This table provides a performance comparison of various text recognition and feature extraction approaches on datasets involving Gujarati text. Traditional methods like HDWT and CCH show varying success, with HDWT achieving moderate accuracy and CCH struggling on structured datasets but performing better on simpler datasets. Deep learning approaches such as LSTM and CNN outperform traditional methods, achieving PCC scores above 98%.

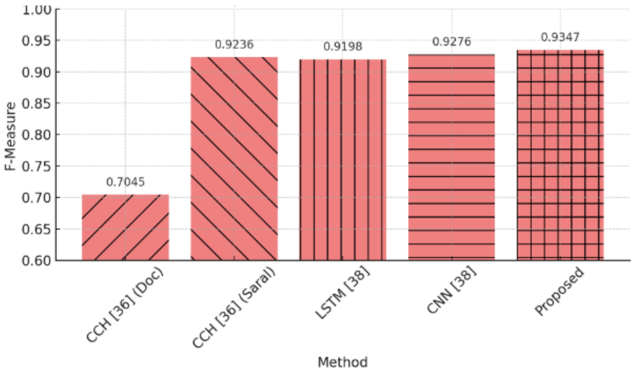
The proposed methods (FRGF ECC and FRGF ECPE) demonstrate strong performance, with PCC values above 92%. These results indicate the effectiveness of the proposed methods for the Gujarati OCR task as compared to another dataset. The different text feature extraction approaches have been compared with the existing one, and a detailed analysis has been presented in the graphs below.



a) Quantitative Precision Analysis: Proposed vs Existing Approaches



b) Quantitative Recall Analysis: Proposed vs Existing Approaches



(c) Quantitative F-Measure Analysis: Proposed vs Existing Approaches

Figure 34: (a)-(c) Quantitative analysis of the proposed method with an existing method with different language datasets (Proposed=FRGF ECC)

5 Conclusion and future scope

This study introduced effective methods for feature extraction of Gujarati text, leveraging fundamental image processing concepts. Given the inherent complexity of the Gujarati language, the research concentrated on the diverse variations of Gujarati text, encompassing 385 distinct characters, including matra's and alphabets. Three specific methodologies were proposed: The Numeric Encoding Approach (ISNEA), The ISNEA model is designed to enhance explainability. Using spatial transitions and direction-based encoding provides complete transparency regarding the decision-making process. This feature fosters trust in the model and improves debugging capabilities, rendering ISNEA appropriate for critical applications, including document OCR in regional languages such as Gujarati. The One-Bit Frequency Count Approach (OBFCA). The OBFCA approach is completely interpretable and traceable. It eliminates the ambiguous latent representations often found in deep learning models, utilizing regionally derived binary structural features that can be plotted and analyzed. This clarity promotes explainability and simplifies debugging in real-world OCR applications, especially for Indian scripts like Gujarati. The FRGF framework links effective feature representation with the interpretability of models. Its architecture ensures that each feature is: Mathematically defined, visually verifiable, and semantically meaningful. This results in notable advantages in explainability over deep neural networks and promotes greater use in real-world OCR systems that demand high accountability, especially for low-resource scripts like Gujarati. These techniques successfully generated features for Gujarati text, which were subsequently catalogued in a Gujarati Text Feature vector dictionary.

A quantitative performance analysis of the proposed methods was conducted, with feature set values illustrated in graphical form for 60 samples selected from a total of 23,100. The experimental results were assessed using the Gujarati OCR dataset and compared against existing methodologies. As part of ongoing research, efforts continue to refine state-of-the-art techniques to achieve optimal results in extracting Gujarati text features. Notably, the FRGF method demonstrated exceptional performance, achieving accuracies of 93.50% and 92.75% for geometric features such as eccentricity and the combined features of perimeter and eccentricity, respectively. However, a limitation identified in the Numeric Encoding Approach is its inability to generate numeric encodings for text containing two or more regions, resulting in only a partial numeric code for the first connected region. To address this limitation, two additional approaches have been proposed and successfully implemented on the Gujarati OCR dataset.

The F1-scores from five cross-validation folds for ECC, AREC, ECLE, and ECPE were subjected to a One-Way ANOVA test in order to assess the statistical significance of performance differences across the feature extraction techniques. With a p-value of 2.04×10^{-18}, the resulting F-statistic was 1003.11, well below the standard cutoff of 0.05. This result validates that the mean F1-scores of the approaches differ statistically significantly. Put differently, not all feature techniques exhibit the same level of performance; at least one, most notably Eccentricity (ECC), exhibits a markedly different and better performance. This statistical confirmation demonstrates that feature extraction strategy selection has a significant influence on OCR accuracy and is not caused by chance variation between folds.

Managing Complex Script Structures: The Gujarati script, like the Devanagari and Bengali scripts, has connected characters, ligatures, and matras that make it more difficult to distinguish and segment characters. In order to overcome these problems, our ISNEA method creates structural representations based on chain codes, a strategy that has been shown effective for Indian scripts in the past (Desai, 2010; Bharvad et al., 2021).

As the Gujarati OCR dataset does not cover the various font difficulties found in public OCR datasets like as ISGUHR (Isolated Gujarati Handwritten Dataset) and Devanagari Character Datasets (DHCD) dataset, nor does it represent real-world handwritten variants, despite having a varied range of character classes and augmentations. Although FRGF achieves a high accuracy 93.5%.

For the validation of the proposed approaches for future work, applying these approaches to noisy, public, and real-world datasets is essential. The future work may also extend to the detection of Gujarati numerals, exploration of additional geometric features using Image processing. The state-of-the-art approaches can also be applied to the Gujarati OCR dataset to enhance text feature extraction performance.

Acknowledgements: Not applicable.

Authors' contributions: The manuscript has been written by author 1 and reviewed by author 2.

Funding: Not applicable.

Availability of data and materials. Data are easily available online.

Declarations

Conflict of interests: The authors declare that they have no competing interests.

Consent for publication: Not applicable.

Ethics approval: Not applicable.

Consent to participate: Not applicable.

References

[1] "invideo.in". Video Marketing Statistics. Accessed on August 4, Available: https://invideo.io/blog/video-marketing-statistics/ 2022.

[2] Dr. A. Anushya. Video Tagging Using Deep Learning: A Survey. International Journal of Computer Science and Mobile Computing, 9(2), 2020.

[3] Hong Liang Xiao Sun, Yunlei Sun, and Yuan Gao. Text feature extraction based on deep learning: a review. EURASIP Journal on Wireless Communications and Networking, Article Number: 211, 2017. https://doi.org/10.1186/s13638-017-0993-1.

[4] S. A. Rajesh, V. A. Bharadi, and P. Jangid. Performance improvement of complex plane-based feature vector for online signature recognition using soft biometric features. International Conference on Communication, Information & Computing Technology, pp. 1–7, 2015. https://doi.org/10.1109/iccict.2015.7045719.

[5] Shah A. and Mishra D. A Review of Biometrics Modalities and Data Mining Algorithm. In Ambient Communications and Computer Systems, Advances in Intelligent Systems and Computing, vol. 696. Springer, 2018. https://doi.org/10.1007/978-981-10-7386-1_51.

[6] N.P. Desai and V.K. Dabhi. Resources and components for Gujarati NLP systems: a survey. Artificial Intelligence Review, 55(11):1–19, 2022. https://doi.org/10.1007/s10462-021-10120-1

[7] J. Bharvad, D. Garg, and S. Ribadiya. A Roadmap on Handwritten Gujarati Digit Recognition using Machine Learning. International Conference for Convergence in Technology, pp. 1–4, 2021. https://doi.org/10.1109/i2ct51068.2021.9418121

[8] K. S. Gautam and M. M. Goswami. Survey on Handwritten Gujarati Word Image Matching. International Conference on Advanced Computing and Communication Systems, pp. 534–538, 2020. https://doi.org/10.1109/icaccs48705.2020.9074274

[9] K. B. Khushali, M. M. Goswami, and S. K. Mitra. Handwritten Gujarati Word Image Matching using Autoencoder. International Conference on Computational Intelligence and Networks, pp. 1–4, 2020. https://doi.org/10.1109/cine48825.2020.234397

[10] P. Jyoti, S. Dimple, K. Rashmi Rekha, and P. Suchit. Gujarati Handwritten Character Recognition from Text Images. Procedia Computer Science, 171, 2020. https://doi.org/10.1016/j.procs.2020.04.055

[11] Apurva Desai. Gujarati handwritten numeral optical character recognition through neural network. Pattern Recognition, 43:2582–2589, 2010. https://doi.org/10.1016/j.patcog.2010.01.008

[12] K. Moro, M. Fakir, B. Bouikhalene, R. El Yachi, and B. El Kessab. Comparison of two feature extraction methods based on the raw form and its skeleton for Gujarati handwritten digits. Facta Universitatis-Series: Mathematics and Informatics, 28(2):161–178, 2013.

[13] K. Moro, M. Fakir, M. Belaid Bouikhalene, B. El Kessab, and R. Ayachi. Gujarati Handwritten Numeral Optical Character through Naive Bayes Classifier. International Journal of Computer Science & Network Solutions, 2, 2014.

[14] P. Goel and A. Ganatra. A Pre-Trained CNN based framework for Handwritten Gujarati Digit Classification using Transfer Learning Approach. International Conference on Smart Systems and Inventive Technology, pp. 1655–1658, 2022. https://doi.org/10.1109/icssit53264.2022.9716483

[15] A. Vanani, V. Patel, K. Limbachiya, and A. Sharma. Handwritten Gujarati Numeral Recognition using Deep Learning. International Conference on Innovative Sustainable Computational Technologies, pp. 1–4, 2022. https://doi.org/10.1109/cisct55310.2022.10046543

[16] Shirke, N. Gaonkar, P. Pandit, and K. Parab. Handwritten Gujarati Script Recognition. International Conference on Advanced Computing and Communication Systems, pp. 1174–1179,

2021.https://doi.org/10.1109/icaccs51430.2021.944 1811

[17] Rajyagor and R. Rakholia. Isolated Gujarati Handwritten Character Recognition using Deep Learning. IEEE International Conference on Electrical, Computer and Communication Technologies, pp. 1–6, 2021. https://doi.org/10.1109/icecct52121.2021.9616652

[18] S. Aniket, R. Atharva, C. Prabha, D. Rupali, and P. Shubham. Handwritten Gujarati script recognition with image processing and deep learning. International Conference on Nascent Technologies in Engineering, pp. 1–4, 2019. https://doi.org/10.1109/icnte44896.2019.8946074

[19] S. Joshi and Y. R. Risodkar. Deep Learning Based Gujarati Handwritten Character Recognition. International Conference On Advances in Communication and Computing Technology, pp. 563–566, 2018. https://doi.org/10.1109/icacct.2018.8529410

[20] B. Divya, M. M. Goswami, and S. Mitra. DNN based approaches for Segmentation of Handwritten Gujarati Text. IEEE International Symposium on Sustainable Energy Signal Processing and Cyber Security, pp. 1–6, 2020. https://doi.org/10.1109/isssc50941.2020.9358904

[21] Abhinav Sharma, Dhiren Soneji, Aabha Ranade, Dhwani Serai, Priya RL, CS Lifna, and Shashikant R. Dugad. Gujarati Script Recognition. Procedia Computer Science, 218, 2023. https://doi.org/10.1016/j.procs.2023.01.204

[22] P. Borad, P. Dethaliya, and A. Mehta. Augmentation based Convolutional Neural Network for recognition of Handwritten Gujarati Characters. IEEE International Conference for Innovation in Technology, pp. 1–4, 2020. https://doi.org/10.1109/inocon50539.2020.9298192

[23] S. Antani and L. Agnihotri. Gujarati character recognition. Proceedings of the International Conference on Document Analysis and Recognition, pp. 418–422, 1999. https://doi.org/10.1109/icdar.1999.791813

[24] M. I. Shah and C. Y. Suen. Word Spotting in Gray Scale Handwritten Pashto Documents. International Conference on Frontiers in Handwriting Recognition, pp. 136–141, 2010. https://doi.org/10.1109/icfhr.2010.28

[25] Kolcz A., J. Alspector, M. Augusteijn, et al. A Line-Oriented Approach to Word Spotting in Handwritten Documents. Pattern Analysis & Applications, pp. 153–168, 2000. https://doi.org/10.1007/s100440070020

[26] Enver Akbacak and Cabir Vural. Deep multi-query video retrieval. Journal of Visual Communication and Image Representation, 85, 2022. https://doi.org/10.1016/j.jvcir.2022.103501

[27] Y. Tewari, P. Soni, S. Singh, M. S. Turlapati, and A. Bhuva. Real Time Sign Language Recognition Framework for Two Way Communication. International Conference on Communication, Information and Computing Technology, pp. 1–6, 2021.https://doi.org/10.1109/iccict50803.2021.9510 094

[28] K. Moro, M. Fakir, B. Bouikhalene, R. El Yachi, and B. El Kessab. New Approach of Feature Extraction Method Based on the Raw form and his Skeleton for Gujarati Handwritten Digits Using Neural Networks Classifier. International Journal of Computer Science & Network Solutions, 2, 2014.

[29] H. Althobaiti and Chao Lu. A survey on Arabic Optical Character Recognition and isolated handwritten Arabic Character Recognition algorithm using encoded freeman chain code. Annual Conference on Information Science and Systems, pp. 1–6, 2017. https://doi.org/10.1109/ciss.2017.7926062

[30] M. A. Mohamad, Z. Musa, and A. R. Ismail. Crow Search Freeman Chain Code Feature Extraction Algorithm for Handwritten Character Recognition. IEEE International Conference On Software Engineering and Computer Systems, pp. 258–263, 2023.https://doi.org/10.1109/icsecs58457.2023.1025 6286

[31] O. Salau and S. Jain. Feature Extraction: A Survey of the Types, Techniques, Applications. International Conference on Signal Processing and Communication, pp. 158–164, 2019. https://doi.org/10.1109/icsc45622.2019.8938371

[32] Rathod Anand, Hasan Mosin, Swadas Prashan, and Desai Nidhi. Gujarati OCR / Typed Gujarati Characters Dataset. CC BY-NC-SA, 2022. https://www.kaggle.com/datasets/ananddd/gujarati-ocr-typed-gujarati-characters

[33] Jagin Patel and Apurva Desai. Gujarati Text Localization, Extraction and Binarization from Images. International Journal of Computer Sciences and Engineering, 6(8):714–724, 2018. https://doi.org/10.26438/ijcse/v6i8.714724

[34] Ami Mehta and Ashish Gor. Multifont multisize Gujarati OCR with style identification. Proceedings of ICECDS, pp. 275–281, 2017. https://doi.org/10.1109/ICECDS.2017.8389951

[35] Ankit Sharma, Priyank Thakkar, Dipak Adhyaru, and Tanish Zaveri. Handwritten Gujarati Character Recognition Using Structural Decomposition

Technique. Pattern Recognition and Image Analysis, 29:325–338, 2019. https://doi.org/10.1134/S1054661819010061

[36] S.B. Suthar and A.R. Thakkar. Dataset Generation for Gujarati Language Using Handwritten Character Images. Wireless Personal Communications, 136:2163–2184, 2024. https://doi.org/10.1007/s11277-024-11369-9

[37] S. Acharya, A.K. Pant, and P.K. Gyawali. Deep Learning based large scale handwritten Devanagari Character Recognition. Proceedings of the 9th International Conference on Software, Knowledge, Information Management & Applications. In press.

[38] Archana Vyas and Swital. Classification of offline Gujarati handwritten characters. Proceedings of ICACCI, 2015. https://doi.org/10.1109/ICACCI.2015.7275831

[39] P. Stefanovic and O. Kurasova. Approach for Multi-Label Text Data Class Verification and Adjustment Based on Self-Organizing Map and Latent Semantic Analysis. Informatica, 33(1):109–130, 2022. https://doi.org/10.15388/22-INFOR473

[40] Laukaitis A, D. Plikynas, and E. Ostasius. Sentence Level Alignment of Digitized Books Parallel Corpora. Informatica, 29(4):693–710, 2019. https://doi.org/10.3233/INF-2018-1200