Gujarati Handwritten Character Recognition Using Hybrid Method Based On Binary Tree-Classifier And K-Nearest Neighbour

Chhaya Patel
MCA Department
Anand Institute of Information Science ,
Anand, India

Apurva Desai
Department of Computer Science,
Veer Narmad South Gujarat University,
Surat, India

Abstract

Gujarati is a language used by more than 50 million people worldwide. Due to dissemination of ICT in India need for Optical Character Recognition (OCR) activities for Indian script is in demand. One can obtain very less OCR related research work for Gujarati script, especially for handwritten form. This paper describes a hybrid approach based on tree classifier and k-Nearest Neighbor (k-NN) for recognition of handwritten Gujarati characters. Combination of structural features and statistical features is used for classification and identification of characters. The features are relatively simple to derive. The structural features are selected by studying the appearance of various handwritten characters. The moment based and centroid based features are first time combined for character recognition of Gujarati script. A success rate of 63% is achieved using proposed method, which is acceptable, as it is one of the few attempts to recognize whole character set of Gujarati handwritten characters.

Keywords- *Feature representation, k-Nearest Neighbor, Moments, Optical Character Recognition (OCR), Tree-classifier*

1. Introduction.

Gujarati script is derived from the popular Devanagri script. The Gujarati language is popularly used by more than 50 million peoples mainly by Gujarati people in the state of Gujarat from India and worldwide as Gujarati people are domicile of many countries. It is found that the work related to Optical Character Recognition (OCR) for Gujarati script is very limited. One can find very few attempts addressing either one or two stages of the OCR phases [10] or OCR for limited characters [9,11] of Gujarati script. It is also observed that majority of work for this language is for printed form rather than handwritten form. A rich cultural heritage is available in handwritten form for this script. Being official language of state of Gujarat major correspondence within various Government departments and other institutes is carried out using Gujarati, either in handwritten or printed form. Many OCR solutions are available for the other languages of Indian origin like Bangala, Devnagri, Gurumukhi but OCR solution for Gujarati handwritten form is not available. The optical character recognition of Gujarati script will definitely be helpful for developing a full-fledged OCR system for Gujarati.

This paper describes an approach to identify numerals and characters of Gujarati script. The suggested approach is based on the structural and statistical features. The structural features are derived based on characteristics of the characters. These features are used to generated a tree classifier that classifies the whole set of characters into subsets. Additional structural features and statistical features are used with k-NN to recognize individual character from each subset at later stage. An overall recognition accuracy of 63.1% is achieved for writer independent data set of characters. The data set is generated by collecting samples from more than 200 different writers of different age group and gender.

2. Characteristics of Gujarati script and challenges for Gujarati OCR

The lack of OCR activities for Gujarati characters may be due to many reasons. The Gujarati script has wide range of characters, which includes - 35 consonants, 13 vowels and 6 signs, 13 dependent vowel signs, 4 additional vowels for Sanskrit, 9 digits and 1 currency sign, which is almost double sized compared to number of alphabets of English language. Fig. 1 shows most frequently used consonants, vowels and numerals.

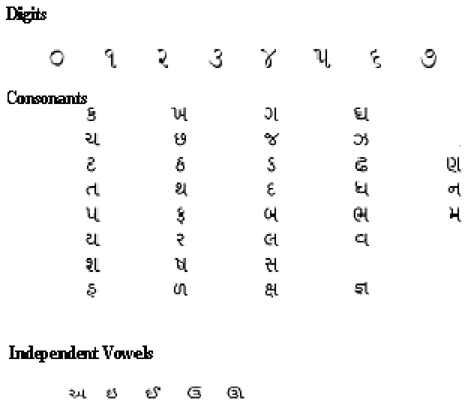


Fig. 1 - The most frequently used consonants, vowels and numerals of Gujarati Script

One can observe that there are many characters that look quite similar to each other. Some of the alphabets resemble the numerical digits, this may create confusion during identification, e.g. the numeral 2-'૨' and the alphabet 'ર', the numeral 4- '૪' and the alphabet 'જ', the numeral 5-'૫' and alphabet 'પ', the numeral '૬' and alphabets 'ફ' and 'ક'. Also the numeral '૯' can easily cause confusion with part of alphabet 'ધ'.

Some characters are quite confusing many times especially in presence of noise that may lead to classification error, even a human may need help of context knowledge to correctly identify them. The characters ફ ડ ર્ક ર્ ઠ forms one such group of confusing characters. Some other such pairs are 'પ' and 'ય' (for handwritten character), 'ઈ' and 'ઇ', 'ઘ' and 'ધ'.

The numeral '૯' and the characters 'ગ', 'ધ', 'ણ', 'શ' are formed using more than one object or parts. Such characters can be easily misinterpreted as sub-part of some characters or conjunct characters that are formed by combining more than one basic character. These characteristics of Gujarati script can be considered as some of the reasons for slow progress of Gujarati OCR activities.

3. OCR activities for handwritten documents in Indian Languages

The publications related to Indian script OCR are less compared to other foreign languages like English, Japanese, and Chinese etc. Indian scripts received attention little later than the other popular languages. Irrespective of any language of Indian origin very few publications are found related to handwritten OCR. It is found that except Hindi, Bangla and few south Indian languages the OCR activities related to handwritten form are negligible.

An off-line recognition of pre-segmented Malayalam handwritten characters based on Kolmogrov-Sminov statistical classifier and k-NN classifier is described in [1]. To identify off-line Devnagri handwritten characters features based on the directional chain code information of the contour points of the characters are suggested in [2] an accuracy of 98.86% and 80.36% is reported for Devnagari numerals and characters respectively. Fuzzy model based recognition of handwritten Hindi numerals is proposed in [3]. The recognition is based on the modified exponential membership function fitted to the fuzzy sets derived from features consisting of normalized distances.

A zone and distance metric based feature extraction system is described in [4] for classification and recognition of Kannada and Telugu script numerals using a centroid based approach. A recognition rate of 98% for Kannada and 86% for Telugu numerals is achieved. Work on recognition of isolated Bangla alphanumeric handwritten characters using a two stage feed forward neural network, trained by back-propagation algorithm is used for recognition in [5]. Another neural network based approach for the recognition of Bangla handwritten numerals is described in [6]. An attempt to recognize Bangla characters is reported in [7].

A system for off-line unconstrained Oriya handwritten numerals is presented in [8]. Histograms of direction chain code of the contour points of the numerals are used as features. A neural network based classifier has been developed supporting an accuracy of 94.81%.

An OCR system for handwritten Gujarati numerals is discussed in [9]. Here in this work a neural network is proposed for identification of Gujarati handwritten digits. A multi layered feed forward neural network is suggested for classification of digits. The features of Gujarati digits are abstracted by four different profiles of digits. Thinning and skew-correction are also done for preprocessing of handwritten numerals before their classification. This work has achieved approximately 82% of success rate for Gujarati handwritten digit identification. Another approach based on hybrid feature extraction technique by same author is suggested in [11]. The structural and statistical features are used for identification of the numerals. An overall accuracy of 96.99% for handwritten Gujarati Numerals is achieved using k-NN as a classifier.

4. Data set generation for Gujarati handwritten characters

There does not exist ready to use data set for Gujarati OCR as it is one of the first few attempts to recognize handwritten Gujarati characters. The author has collected samples of all characters of Gujarati script from more than 200 different writers of different age group and gender to form the data set. To make the data set stable and usable some of the samples were collected from writers who did not know Gujarati script. Each collected data form was scanned at resolution of 200 to 300 dpi using a flatbed scanner. Collection of individual character was generated by separating them from the input form image. These separated characters were preprocessed at next stage.

5. Preprocessing

The individual handwritten character will be different from the other in a same character group, as they are written by different people using different pens having different ink and different tip size. Following Fig. 2 shows scanned images of the Gujarati numeral 6. One can observe the variations in thickness and size of each sample. Presence of slant due to writing style is frequently found in handwritten form one can notice it from Fig.2. As handwriting of any two persons are hardly same one have totally different image for individual character in data set.

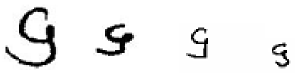


Fig. 2 – Sample images for Gujarati numeral 6

Another such case is shown in Fig. 3 for a Gujarati character ‘ka’ written by different writers. One can observe more variations in size, thickness, slant and formation of the character. It is also noticeable that the letter ‘ka’ and the numeral ‘6’, as shown in Fig. 2 have sufficient similarities to cause confusion while recognizing.

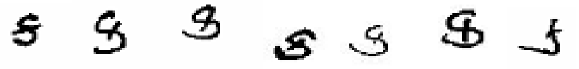


Fig. 3 – Sample images for Gujarati character ‘ka’

A preprocessing is required to convert an input character image into binary form that is noise free, smooth, and thin. Such images preserve the shape information with minimum storage requirements and help to improve accuracy of classification and recognition phase.

During preprocessing phase, first the image contrast is adjusted in order to remove effect of different colored ink and effect of thin pointed tip causing sometimes light appearance of a character. The image intensities are adjusted based on the adaptive histogram equalization algorithm. The noise introduced during digitization is removed by using two-dimensional adaptive Wiener filter. The Wiener filter is a low-pass filter that uses a pixel wise adaptive Wiener method based on statistics estimated from a local neighborhood of each pixel. The Wiener filter estimates the local mean and variance around each pixel. In our case for each pixel each 3X3 neighborhood pixels are used for filtering. The image is then converted into binary form by using threshold value.

An Ostu’s method is used to determine the global image threshold. This global threshold (level) is used to convert an intensity image to a binary image that can be used for feature extractions and recognition. This binary image is cropped to remove unwanted pixels present surrounding the character image. Fig. 4 shows original input character image, binary form of the image and the cropped image.

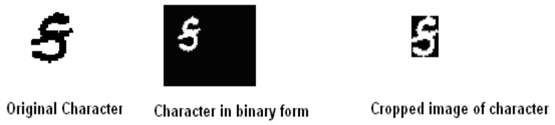


Fig. 4 – The input character and the cropped binary image for the character

All the images of the characters need to be converted into a standard size for further processing. Normalization is carried out for converting each character image into size of 40 X 40 pixel image.

6. Feature extraction

The features are divided in two categories in this case - primary features and secondary features. The primary features are selected based on the characteristics of Gujarati script. These features are reasonably invariant with respect to shape variations caused by various writing styles, easy to implement, fast to generate, size independent and easy to derive. These features are structural features and are used primarily to divide the set of basic characters into smaller manageable subsets using a binary tree classifier and secondarily to recognize a character at later stages.

The binary tree classifiers are one of the popular classifiers. They were introduced nearly twenty five years ago and are being used by many researchers for classification of characters. Many references are available where tree classifiers have been used for OCR activities related to Indian origin scripts. A Gurumukhi script recognition is described in [18] and [22]. Tree classifier is used for printed Oriya script in [19]. A Bangla and Devnagri classifiers are described in [20] and [21]. The features used for recognition of Gujarati characters are described below.

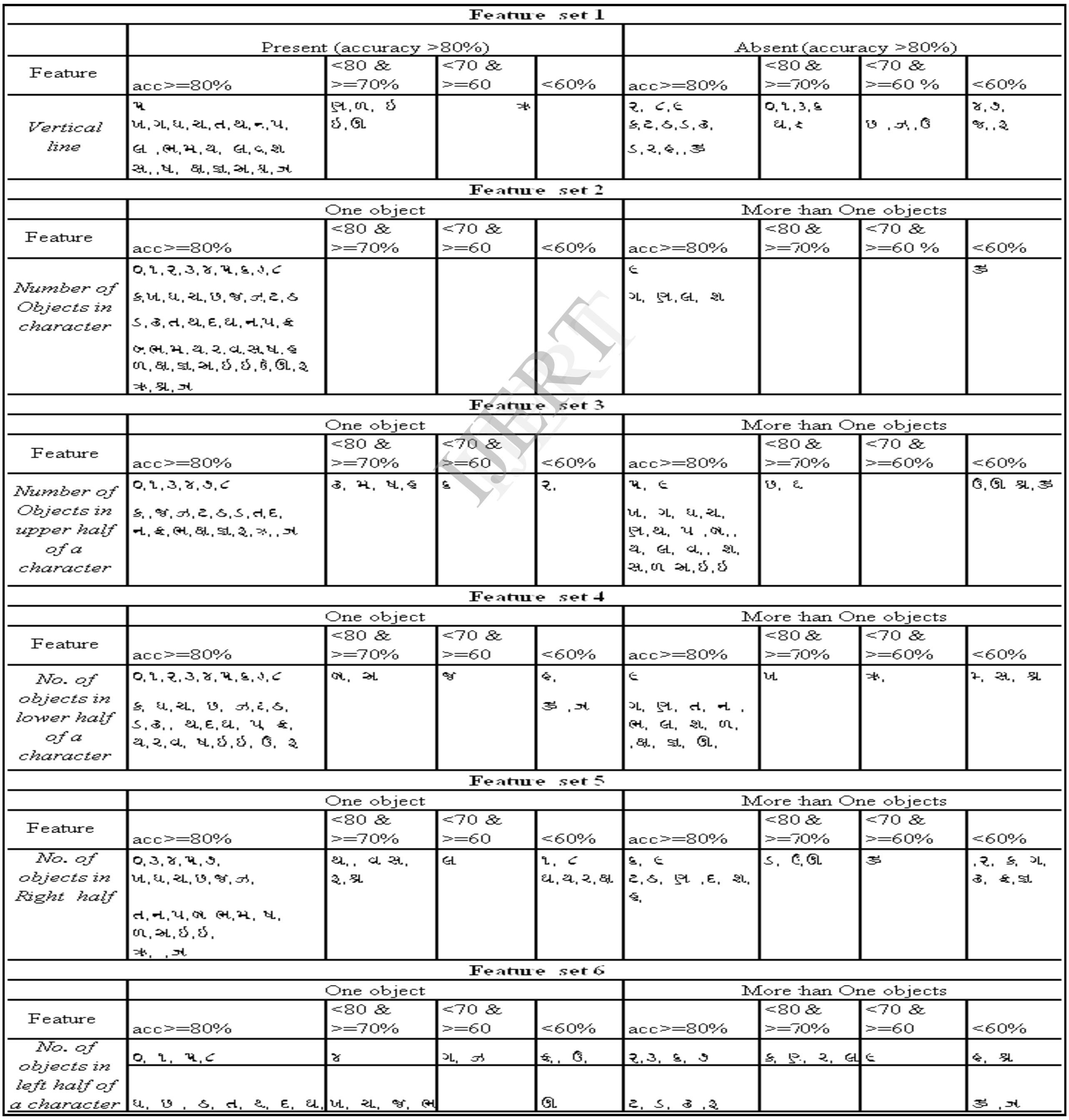
6.1 Primary features and its usage for character classification

The primary features are derived using the structural features of character image. Table-1 describes various features that were been tested for present work. The classification accuracy is divided into four ranges for each feature, namely more than 80%, 70 to 79%, 60 to 69% and below 60%. In table-1 classification accuracies are specifies for various features for each character of the Gujarati script.

The features selected are number of objects in character, number of objects in upper and lower half of the image, number of holes in character image, number of objects in left half of the image and number of objects in right half of the image. It is found that number of objects in left half of the image and number of objects in right half of the image can be ignored as far as usage with tree classifier is considered. The remaining features are well suited as a primary feature to design the tree classifier that divides the character set of Gujarati script into subsets which then can be used for character identification.

This primary features for Gujarati character classification are determined by studying the formation of each characters and writing styles of 210 different writers hence the proposed approach for recognition of Gujarati handwritten characters can be considered as a writer independent approach for classification.

Table 1 – Results of various features used for classification of Gujarati characters



Feature set 1								
	Present (accuracy >80%)				Absent(accuracy >80%)			
Feature	acc>=80%	<80 & >=70%	<70 & >=60	<60%	acc>=80%	<80 & >=70%	<70 & >=60 %	<60%
Vertical line	૫ ખ,ગ,ઘ,ચ,ત,થ,ન,પ, લ ,ભ,મ,ય, ધ,વ,શ સ,,ષ, ક્ષ,જ્ઞ,અ,શ્ર,ત્ર	ઉ,ળ, ઇ ઈ,ઊ	એ		૨, ૮,૯ ક,ટ,ઠ,ડ,ઢ, ડ,ર,હ,,ઞ	૦,૧,૩,૬ ઘ,ર	છ ,ઝ,ઉ	૪,૭, જ,,ઋ
Feature set 2								
	One object				More than One objects			
Feature	acc>=80%	<80 & >=70%	<70 & >=60	<60%	acc>=80%	<80 & >=70%	<70 & >=60 %	<60%
Number of Objects in character	૦,૧,૨,૩,૪,૫,૬,૭,૮ ક,ખ,ઘ,ચ,છ,જ,ઝ,ટ,ઠ ડ,ઢ,ત,થ,દ,ધ,ન,પ,ફ બ,ભ,મ,ય,ર,વ,સ,ષ,હ ળ,ક્ષ,જ્ઞ,અ,ઇ,ઈ,ઉ,ઊ,ઋ એ,શ્ર,ત્ર				૯ ગ, ઉ,લ, શ			ઞ
Feature set 3								
	One object				More than One objects			
Feature	acc>=80%	<80 & >=70%	<70 & >=60	<60%	acc>=80%	<80 & >=70%	<70 & >=60%	<60%
Number of Objects in upper half of a character	૦,૧,૩,૪,૭,૮ ક,,જ,ઝ,ટ,ઠ,ડ,ત,દ, ન,ફ,ભ,ક્ષ,જ્ઞ,ઋ,એ,,ત્ર	ઢ, મ, ષ,હ	૬	૨,	૫, ૯ ખ, ગ, ઘ,ચ, ઉ,થ, પ ,બ,, ય, લ, વ,, શ, સ,ળ અ,ઇ,ઈ	છ, દ		ઉ,ઊ શ્ર,ઞ
Feature set 4								
	One object				More than One objects			
Feature	acc>=80%	<80 & >=70%	<70 & >=60	<60%	acc>=80%	<80 & >=70%	<70 & >=60%	<60%
No. of objects in lower half of a character	૦,૧,૨,૩,૪,૫,૬,૭,૮ ક, ઘ,ચ, છ, ઝ,ટ,ઠ, ડ,ઢ,, થ,દ,ધ, પ ફ, ય,ર,વ, ષ,ઇ,ઈ, ઉ, ઋ	બ, અ	જ	હ, ઞ ,ત્ર	૯ ગ, ઉ, ત, ન , ભ, લ, શ, ળ, ,ક્ષ, જ્ઞ, ઊ,	ખ	એ,	મ, સ, શ્ર
Feature set 5								
	One object				More than One objects			
Feature	acc>=80%	<80 & >=70%	<70 & >=60	<60%	acc>=80%	<80 & >=70%	<70 & >=60%	<60%
No. of objects in Right half	૦,૩,૪,૫,૭, ખ,ઘ,ચ,છ,જ,ઝ, ત,ન,પ,બ ભ,મ, ષ, ળ,અ,ઇ,ઈ, એ, ,ત્ર	થ,, વ સ, ઋ,શ્ર	લ	૧, ૮ ધ,ય,ર,ક્ષ	૬, ૯ ટ,ઠ, ઉ ,દ, શ, હ,	ડ, ઉ,ઊ	ઞ	,૨, ક, ગ, ઢ, ફ,જ્ઞ
Feature set 6								
	One object				More than One objects			
Feature	acc>=80%	<80 & >=70%	<70 & >=60	<60%	acc>=80%	<80 & >=70%	<70 & >=60	<60%
No. of objects in left half of a character	૦, ૧, ૫,૮	૪	ગ, ઝ	ક,, ઉ,	૨,૩, ૬, ૭	ક, ઉ,, ર, લ	૯	હ, શ્ર
	ઘ, છ , ઠ, ત, દ, દ, ધ,	ખ, ચ, જ, ભ		ઊ	ટ, ડ, ઢ ,ઋ			ઞ ,ત્ર

The primary features used with tree classifiers are specified below:

Number of objects in the character (1 or >1)

Number of objects in upper half of a character (1 or >1)

Number of objects in lower half of a character (1 or >1)

Number of holes into a character (0 or >0)

Each of these features is used to generate a binary value that can be used to divide the character sets of Gujarati into small subsets of few characters as shown in Fig 5.

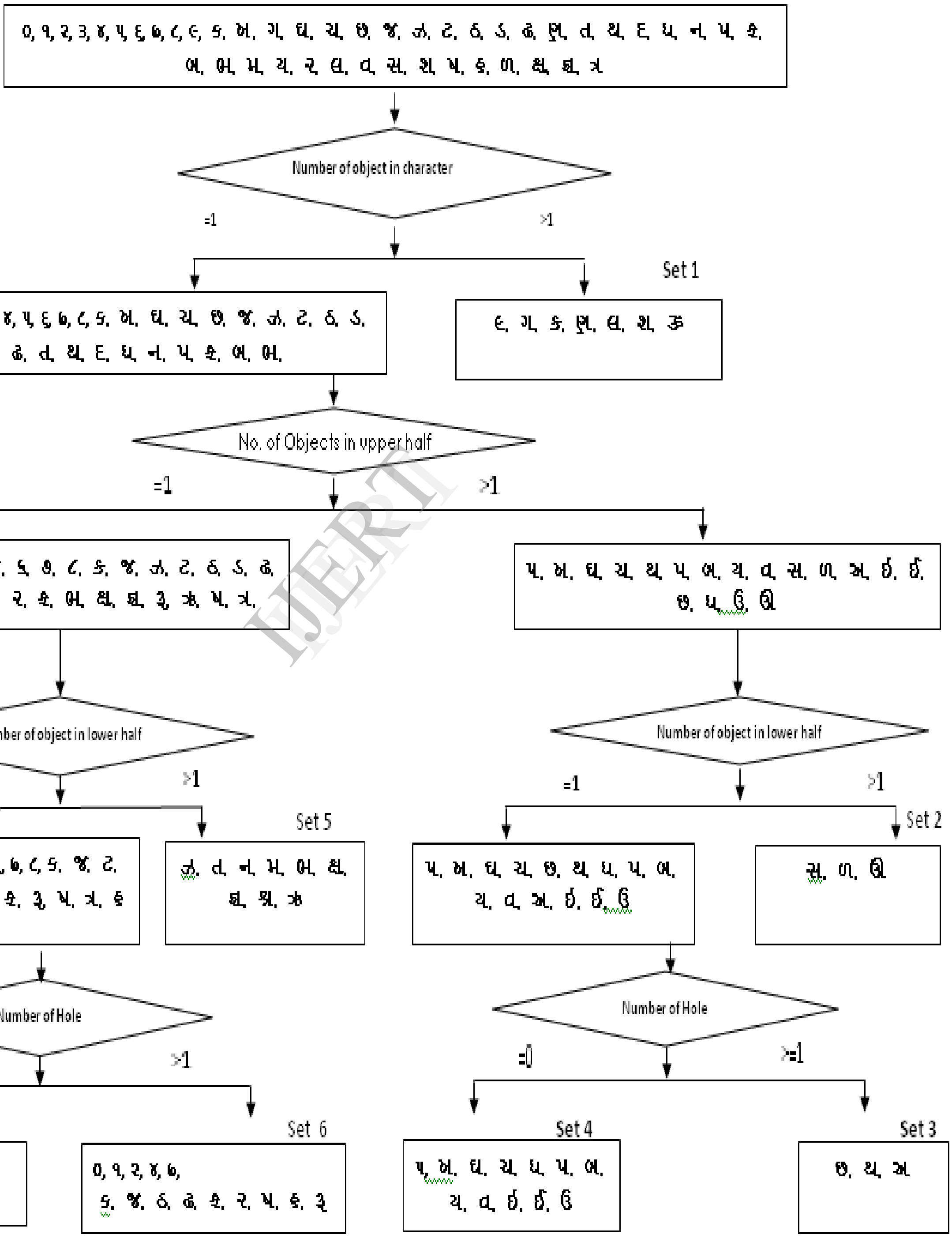

Number of object in character
=1
>1
Set 1
No. of Objects in upper half
=1
>1
Number of object in lower half
=1
>1
Set 5
Number of object in lower half
=1
>1
Set 2
Number of Hole
=0
>1
Set 7
Set 6
Number of Hole
=0
>=1
Set 4
Set 3

Fig. 5 – Proposed Tree classifier for Gujarati basic characters

One can observe that the first feature is number of components (objects) used to form a character. The labeling approach is used to determine number of objects or parts used to form a character, e.g. in numerical digits of Gujarati script only the number nine(૯) has more than one objects used to form a digit. Similarly the characters- ચ, ણ, ધ, શ, ઝ have more than one objects used to form a character. The remaining characters have connected objects or single object. A set with the characters ૯, ચ, ણ, ધ, શ, ઝ is named as Set1 which can be used for further recognition. The remaining characters are further classified using different features at subsequent levels.

The next level subdivides the remaining characters based on the number of components in upper half of the character image. As we can see from Fig. 5 that it further divides the collection of characters into small sets. The number of objects in lower half of a character image is considered for further subdivision of the character set, generating Set2, and a collection of other characters which can be finally sub divided using another strong feature namely presence of hole in a character. Using this approach character set is subdivided and classified into total seven subsets as shown in Fig. 5. These subsets are small enough and hence can be managed easily.

The application of binary tree based classifier has its own advantages and disadvantages. The binary tree classifier provides speed but, they are sensitive to noise and fonts. The decisions may be wrong if features are not derived properly especially for noisy characters. Hence in our case combinations of strong features which have noise resistance have been used. The tree based approach is used only to derive subsets of characters and not for recognition of character, thus reducing probability of incorrect recognition.

6.2 Secondary features

Keeping in view the advantages and disadvantages of binary classifier a combination of binary classifier and k-Nearest Neighbor has been suggested. Once the character is classified to fall into a particular subset, some additional features / secondary features are derived, based on which final identification is carried out. The secondary features used in this study are described below.

i) **The averaging feature:** This feature is derived by application of averaging principal on the binary image. The original image of size 40 X 40 is subdivided into 4X4 blocks and average ON pixel values in each block is used to form a feature. The image blocks are extracted as shown in Fig. 6 , in row major order.

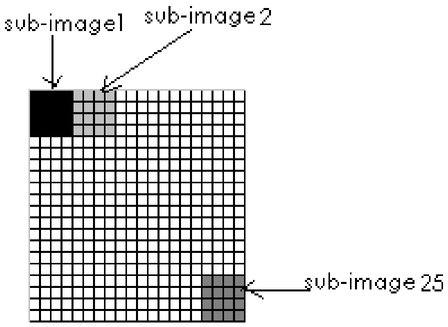


Fig. 6 – Computing the average value of ON pixels in image

ii) **The Moment based features:** The third feature is derived using moments as moments are capable to describe shape's layout (the arrangement of its pixels). Moments provides global description of a shape, accruing this same advantage as Fourier descriptors since there is an in-built ability to discern, and filter, noise. According to [23] moments for image analysis were introduced in the 1960s. An excellent and fairly up-to-date review for usage of moments for the same purpose is available in [24].

Centralized moments are well suited for application like character recognition as they are translation invariant. In order to accrue invariance to scale, we require normalized central moments η_{pq} defined as,

$$\text{Where} \quad n_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}}$$

$$\gamma = \frac{p+q}{2} + 1 \qquad \forall p+q \geq 2$$

Region moment representations interpret a normalized gray-level image function as a probability density of a 2D random variable. Assuming that non-zero pixel values represent regions, moments can be used for binary or gray-level transformations. For a digital image the central moments can be expressed as:

$$\mu_{pq} = \sum_{x} \sum_{y} (x - \bar{x})^{p} (y - \bar{y})^{q} f(x, y)$$

where x, y are the co-ordinates of the region's center of gravity (centroid). These can be obtained using the following equations:

$$\bar{x} = \frac{m10}{m00} \quad \text{and} \quad \bar{y} = \frac{m01}{m00}$$

The central moments of up to order 3 can be obtained from the above equation by choosing p, q = 0, 1, 2, 3, such that $p + q \leq 3$.

The normalized central moments denoted by η_{pq}, are denoted by $\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{y}}$ where $y = (p+ q)/2 + 1$ for $p + q = 2, 3....$.

Rotation invariance can be achieved if the coordinate system is chosen such that $\mu 11 = 0$. Seven rotation, translation, and scale invariant moment characteristics can be derived from the second and third moments. The moments values are calculated based on following set of equations.

$M1 = \eta_{20} + \eta_{02}$

$M2 = (\eta_{20} - \eta_{02})^2 + 4\,\eta_{11}^2$

$M3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$

$M4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$

$M5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) + ((\eta_{30} + \eta_{12})^2 - 3(\eta_{21} - \eta_{03})^2) + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})((3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2)$

$M6 = (\eta_{20} - \eta_{02})((\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2) + 4\,\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$

$M7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})((\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2) + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})((3(\eta_{12} + \eta_{30})^2 - (\eta_{21} + \eta_{03})^2)$

Here M1 and M2, are second-order moments as p + q = 2 for them. The remaining are third-order moments, since p + q = 3. The moment M7 is a skew invariant moment and is used to distinguish mirror images. The seven moments M1,M2….M7 values are used for creating a feature vector for a given character image. The character image of size 40 X 40 is segmented into four equal sub-images of size 20 X 20 each and for each sub-image moment features are calculated. This process produces total 7 X 4=28 different values for each character.

iii) The Centroid distance based features: The fourth feature is derived by using centroid distance function. The thinned, resized and cropped binary image of the character is segmented into 64 equal sub-images. For all ON pixels in each sub-image, the distance from the centroid of the original character image is computed. Using these distance values an average distance value for individual sub-image is computed, i.e. for each sub-image there will be one average distance value if at least one pixel is ON in that sub-image. Thus total 64 average values are computed per character image. This collection of 64 average values is used as one of the feature for character recognition. If some zones are empty there will be no ON pixels, for such zones value of feature vector will be zero.

These secondary features are used to recognize an individual character using k-NN.

7. Character Recognition

To eliminate limitations of binary tree classifier especially for noisy characters, the tree based approach is used only to derive subsets of characters and not for recognition of character, thus reducing probability of incorrect recognition. A hybrid approach based on subdivision of character set into subsets using tree classifier and recognition using k-NN is developed for Gujarati handwritten OCR.

The feature vector that is used to recognize a character is a combination of various features discussed above. It is formed using 5 elements that are based on structural features namely objects in character, objects in left half , right half , lower half and upper half of the characters, 28 elements based on invariant moments, 25 elements based on average pixels and 64 elements are based on centroid distance. This feature vector is used for character recognition, thus a feature string of total 122 elements is used to recognize an individual character.

7.1 Recognition of Gujarati Characters using k-NN

The k-Nearest Neighbor (k-NN) is used for recognition of characters from these derived subsets. k-NN is a method for classifying objects based on closest training examples in the feature space. Here an object is classified by a majority vote of its neighbor, with the object being assigned to the class most common amongst its k - nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of its nearest neighbor. The neighbors are taken from a set of objects for which the correct classification is known. This can be thought of as the training set for the algorithm.

Thus k-NN is an instance based classification algorithm. Many researchers have found that the k-NN algorithm achieves very good performance for character recognition in their experiments on different data sets. The idea behind k-Nearest Neighbor algorithm is quite straightforward. To classify a new character, the system finds the k -nearest neighbors among the training datasets, and uses the categories of the k- nearest neighbors to weight the category candidates. The k-NN algorithm can be described using the following equation (1)

$$y(d_i) = \arg\max_k \sum_{x_j \in kNN} Sim(d_i, x_j)\, y(x_j, c_k) \quad .... (1)$$

where d_i is a test character, x_j is one of the neighbors in the training set, $y(x_j,c_k) \in \{0,1\}$indicates whether x_j belongs to class c_k, and $Sim(d_i,x_j)$ is the similarity function for d_i. Equation (1) means the class with maximal sum of similarity will be the winner.

The k-NN algorithm measures the distance between a query scenario and a set of scenarios in the data set to determine the similarity. We can compute the distance between two scenarios using some distance function d(x,y), where x , y are scenarios composed of N features, such that x = $\{x_1,x_2,\ldots x_N\}$, y = $\{ y_1, y_2,\ldots y_N\}$. Some of the popular distance measures are, Absolute distance, Euclidean Distance, Hamming distance, Mahalanobis distance. The selection of suitable distance measure function and value of parameter k is very important factor deciding the performance of the k-NN classifier.

Here we have used the Euclidean distance measure for k-NN. The Euclidean distance is given by equation (2)

$$d_E(x,y) = \sum_{i=1}^{N} \sqrt{x_i^2 - y_i^2} \quad \text{…. (2)}$$

This equation is used to measure the distance between two scenarios. One can simply pass through the data set, one scenario at a time, and compare it to the query scenario.

The data sets can be represented as a matrix D = N X P, containing P scenarios $s_1,s_2,..,s_p$, where each scenario s_t contains N features $s_t= \{s_{1i},\ldots s_{Ni}\}$. A vector O with length P of output values O= $\{o_1, o_2,\ldots o_P\}$ accompanies this matrix, listing the output ot for each scenario st.

A detailed discussion on the k-NN classification technique can be found in [12]. The review of literature suggests that k-NN is widely used for character recognition for languages of Indian origin [13,14,15], as well as other languages. According to [16] k-NN classier has consistently outperformed by the other classifiers for handwritten digit recognition. A combination of fuzzy k-NN and k-NN is suggested in [17] for recognition of handwritten Kannada numerals.

Following algorithm states the hybrid approach using Tree-classifier and k-NN to recognize a Gujarati character.

Algorithm Hybrid recognizer:

i) Fetch the character to be recognized.
ii) Generate the preliminary feature set for the character.
iii) Using the preliminary features, determine in which subset the character belong to. Accordingly set the subset flag indicating the subset number.
iv) Generate the secondary feature set for the character.
v) Using the subset flag determined in step 4, select the training set for k-NN, use k-NN for identification of character.
vi) Display the unique code of the identified character.

The proposed method has provided an accuracy of 63.1% for given data set.

8. Results

The results are shown in Table -2. As it is the first attempt to identify the handwritten basic characters of the Gujarati script the accuracy is acceptable. Some of the reasons for less accuracy are, similarity of some characters like the digit '5'-૫ and the basic character 'pa'-પ .The number 2-૨ and the character 'ra'- ર, the characters i'-ઇ and 'i_bar'-ઈ , characters 'dha'- ધ and 'gha'- ઘ, the number '4'- ૪ and character 'ja'- જ. These confusing characters cause reduction in recognition accuracy.

Table 2 – Results of k-NN for handwritten Gujarati characters recognition

Set No.	Characters														SET wise Average accuracy %
Set1	૯	ગ	ણ	ઘ	શ	ક									69.8
% Accuracy for individual character	87	64	51	57	61	97									
Set2	સ	ળ	ઊ												62.9
% Accuracy for individual character	66	54	69												
Set3	ઙ	થ	અ												79.5
% Accuracy for individual character	80	60	99												
Set4	ય	ષ	ધ	ચ	ઘ	પ	ભ	વ	લ	ઇ	ઈ	ઉ			55.8
% Accuracy for individual character	100	47	21	47	36	86	57	26	23	70	91	66			
Set5	ઝ	ત	ન	મ	ભ	ક્ષ	ડ	શ્ર	જ્ઞ						45.7
% Accuracy for individual character	20	46	40	37	54	29	41	56	89						
Set6	૦	૧	૨	૪	૭	ક	જ	ઠ	ઢ	ફ	ર	૫	હ	રૂ	54.7
% Accuracy for individual character	99	84	93	21	87	49	54	24	39	37	59	63	1	56	
Set7	૩	૬	૮	ટ	ડ	દ									73.3
% Accuracy for individual character	84	46	99	79	64	69									
														Overall Accuracy	**63.1**

9. Conclusions

It is one of the few attempts to address the issue of Optical Character Recognition for Gujarati handwritten characters. The structural features selected for recognition are easy to obtain and are used for the first time for Gujarati script. The other statistical features are also applied first time for Gujarati script. While designing algorithms, factors like – simplicity, ease of implementation and speed were considered to have a fast and accurate solution for an OCR problem. A hybrid approach using binary tree and k-NN is also suggested first time for Gujarati handwritten characters. An overall recognition accuracy of 63.1% for character recognition is obtained. The results are satisfactory, as it is just beginning of this untouched research area compared to results for other Indian scripts in printed and handwritten form. The work hopefully will be a stepping stone for future research work for Gujarati language or similar Indian languages.

10. References

[1] V.L. Lajish, T.T.K. Suneesh and N.K. Narayanan,"Recognition of Isolated Handwritten Character Images using Kolmogorov-smirnov ,Statistical Classifier and K-nearest Neighbour Classifier", Proc. of the International Conference on Cognition and Recognition, pp 526-531, 2005.

[2] N. Sharma, U. Pal,F. Kimura, and S. Pal P. Kalra and S. Peleg (Eds.), "Recognition of Off-Line Handwritten Devnagari Characters Using Quadratic Classifier", ICVGIP 2006, LNCS 4338, Springer-Verlag Berlin Heidelberg, pp. 805-816,2006 .

[3] M. Hanmandlu and O.V. Ramana Murthy, " Fuzzy Model Based Recognition of Handwritten Hindi Numerals", Science Direct, Pattern Recognition Volume 40, Issue 6, Proc. of the International Conference on Cognition and Recognition, pp. 1840-1854, June 2007.

[4] Rajashekararadhya, S.V.; Ranjan, P.V., "Neural network based handwritten numeral recognition of Kannada and Telugu scripts", TENCON 2008,IEEE Region 10 Conference , pp .1 – 5 Nov. 2008.

[5] A. Dutta and S. Chaudhury, "Bengali alpha-numeric character recognition using curvature features", Patter Recognition, Vol. 26(12), pp. 1757-1770, 1993.

[6] U. Bhattacharya, T. K. Das, A. Datta, S. K. Parui and B. Chaudhuri, "A hybrid scheme for handprinted numeral recognition based on a self-organizing network and MLP classifiers", International Journal on Pattern Recognition and Artificial Intelligence, Vol. 16(7), pp. 845-864, 2002.

[7] U. Pal and S. Datta, "Segmentation of Bangla unconstrained handwritten text", Proceedings of 7th ICDAR, pp. 1128-1132, 2003.

[8] K. Roy, T. Pal, U. Pal and F. Kimura, "Oriya handwritten numeral recognition system", Proceedings of ICDAR, pp. 770-774, 2005.

[9] Apurva A. Desai, "Gujarati handwritten numeral optical character reorganization through neural network", Pattern Recognition Volume 43, Issue 7, pp. 2582-2589, July 2010.

[10]Jignesh Dholakia , Atul Negi, S. Rama Mohan, "Zone Identification in the Printed Gujarati Text", Proceedings of Eight International Conference on Document Analysis and Recognition (ICDAR'05), pp.272-276, 2005.

[11]Apurva A.Desai ,"Handwritten Gujarati Numeral Optical Character Recognition using Hybrid Feature Extraction Technique", Proceedings of International Conference on Image processing, computer vision & pattern recognition,IPCV'10. pp. 733-739,2010.

[12]B. V. Dasarathy, "Nearest neighbor pattern classification techniques", IEEE Computer Society Press,New York, 1991.

[13]Anilkumar N. Holambe,Dr.Ravinder.C.Thool, "Printed and Handwritten Character &Number Recognition of Devanagari Script using SVM and KNN",International Journal of Recent Trends in Engineering and Technology,Vol.3, No.2, pp.163-166,2010.

[14]Sanghamitra Mohanty, Himadri Nandini Das Bebartta, "Performance Comparison of SVM and K-NN for Oriya Character Recognition", International Journal of Advanced Computer Science and Applications (IJACSA), Special Issue on Image Processing and Analysis, pp.-112-116, 2011.

[15]B.V.Dhandra, Mallikarjun Hangarge, Gururaj Mukarambi, "Spatial Features for Handwritten Kannada and English Character Recognition", International Journal of Computer Applications,Special Issue on Recent Trends in Image Processing and Pattern Recognition, pp.-146-151, 2010.

[16]Cheng-Lin Liu, Kazuki Nakashima, Hiroshi Sako, Hiromichi Fujisawa, "Handwritten digit recognition: benchmarking of state-of-the-art techniques", Pattern Recognition Vol. 36, pp.2271 – 2285, 2003.

[17]Dinesh Acharya U.,N. V. Subba Reddy, and Krishnamoorthi Makkithaya, "Multilevel Classifiers in Recognition of Handwritten Kannada Numerals", World Academy of Science, Engineering and Technology Vol. 42, pp. 278-283,2008.

[18] G. S. Lehal and Chandan Singh, "A Gurmukhi Script Recognition System", Proceedings of 15th ICPR, Vol. 2, pp. 557-560,2000.

[19] B. B. Chaudhuri and U. Pal and M Mitra, " Automatic recognition of printed Oriya script", S¯adhan¯a Vol. 27, Part 1, pp. 23–34, February 2002.

[20] B. B. Chaudhuri and U. Pal, " A Complete Printed Bangla OCR System" , Pattern Recognition, Vol. 31, pp. 531-549, 1998.

[21] B. B. Chaudhuri and U. Pal, " An OCR System To Read Two Indian Language Scripts: Bangla And Devnagari (Hindi)", Proceedings of Fourth International conferance on Document Analysis and Recognition, IEEE Computer Society Press, pp.1011-1016, 1997.

[22] G S Lehal and Chandan Singh, "A Complete OCR System For Gurmukhi Script", Proceedings of SSPR 2002, Lecture Notes in Computer Science, Vol. 2248, Springer- Verlag, Germany, pp. 344-352, 2002.

[23] Hu, M. K., "Visual Pattern Recognition by Moment Invariants" , IRE Transaction on Information Theory, IT-8, pp. 179–187, 1962.

[24]Prokop, R. J. and Reeves, A. P., "A Survey of Moment-Based Techniques for Unoccluded Object Representation and Recognition", CVGIP: Graphical Models and Image Processing, 54(5), pp. 438–460, 1992.