ISSN (e): 2250 – 3005 || Volume, 06 || Issue, 09|| September – 2016 ||
International Journal of Computational Engineering Research (IJCER)

Character Recognition System for Modi Script

Pankaj A. Patil
S.S.B.T.'s College of Engineering and Technology Bambhori, Jalgaon,

ABSTRACT
Optical character recognition (OCR) is the mechanical or electronic conversion of images of typed, handwritten or printed text into machine-generated text. Character recognition is one of the oldest fields of research since the advent of computers. Till now OCR has been developed for several languages like English, Devnagari (Hindi), Bangla etc. But not much work has been done for Modi script. Modi script has been used in Maharashtra in the 17th century, Modi script was widely used in Maharashtra for writing up to 1950 and contained lot of literature of various philosophers and has its own historical importance. And to convert that literature into computer readable format using optical character recognition systems is our objective. Hence the proposed system will be beneficial for extracting the literature from Modi script. This project work elaborates algorithm required for Modi script OCR. After initial stage of Segmentation, we have used Affine Moment Invariants for calculating moments of each character. These moments were used for training of the system and building database. Thereafter classification has been performed using Fuzzy Logic.

Keywords: *Affine Moments Invariants, Character Recognition, Fuzzy Logic, Modi, OCR, Optical Character Recognition, Segmentation*

I. INTRODUCTION

Machine simulation of human functions has been a very challenging research field since the advent of digital computers. In some areas, which require certain amount of intelligence, such as number crunching or chess playing, tremendous improvements are achieved. On the other hand, humans still outperform even the most powerful computers in the relatively routine functions such as vision. Machine simulation of human reading is one of these areas, which has been the subject of intensive research for the last three decades, yet it is still far from the final frontier.

In the present scenario more importance is given for the "paperless office" there by more and more communication and storage of documents is performed digitally. Documents and files that were once stored physically on paper are now being converted into electronic form in order to facilitate quicker additions, searches, and modifications, as well as to prolong the life of such records. Because of this, there is a great demand for software, which automatically extracts, analyze, recognize and store information from physical documents for later retrieval. One of the important steps of document processing is Textual processing through Optical character recognizer (OCR).

Optical Character Recognition (OCR) is a branch of pattern recognition and computer vision. OCR has been extensively researched for more than four decades. With the advent of digital computers, many researchers and engineers have been engaged in this important area. OCR is broadly defined as the process of recognition either printed or handwritten text from document images and converting into electronic form. It is not only a new developing area due to many potential applications such as bank check processing, postal mail sorting, automatic reading of tax forms, and reading various handwritten and printed text and non-text documents. It is also a benchmark for testing and verifying new pattern recognition theories and algorithms. In recent years, many new classifiers and feature extraction algorithms have been proposed and tested on various OCR databases and these techniques have been used in wide applications. In the last two centuries there have been significant efforts to develop systems which will able to identify the scripts and recognize their contents. The problem of character recognition is becoming more and more important in the modem world; in general and handwritten character recognition system in particular for fast processing of the document images. India is multilingual and multi script country and uses 18 scripts. Officially Indian states are using three scripts viz., English as first language, Hindi as second language and local language of the states as third language. Hence there is a need for an OCR system for multilingual and multi script for an Indian contest OCR system. Thus a development of multilingual OCR system is one of the thrust areas and it has potential contribution for scientific and economy advancement of the country. Numerous scientific papers on OCR have been reported in the literature. Today, research on OCR system is addressing number of diversified sophisticated problems. Important components in OCR includes analysis, recognition of complex documents including texts, images, charts and tables, script recognition, printed/handwritten character recognition, and multilingual character recognition. Handwritten Character Recognition is difficult task, since the variation in handwritten documents depends on writer's age, gender, education, ethnic background as well as the writer's mode while writing. Recognition of character from single script is relatively simpler than the multilingual characters. Multilingual OCR system has wide applications but it has not gained public attention until recent scenario.

II. LITERATURE SURVEY

The Brahmic family of scripts uses Abugida writing system.It originated from the ancient Indian Brahmi script and includes nearly all of the scripts of India and southeast Asia. In Fig. 5, we draw a tree diagram to illustrate the evolution of major Brahmic scripts in India and southeast Asia. The northern group of Brahmic scripts (e.g., Devnagari, Bengali, Manipuri, Gurumukhi, Gujrati, and Oriya) bears a strong resemblance to the original Brahmi script. On the other hand, scripts in south India (Tamil, Telugu, Kannada, and Malayalam) as well as in southeast Asia (e.g., Thai, Lao, Burmese, Javanese, and Balinese) are derived from Brahmi through many changes and so look quite different from the northern group. One important characteristic of Devnagari, Bengali, Gurumukhi, and Manipuri is that the characters in a word are generally written together without spaces so that the top bar is unbroken. This results in the formation of a headline, called shirorekha, at the top of each word. Accordingly, these scripts can be separated from other script types by detecting the presence of a large number of horizontal lines in the textual portions of a document. Modi is a Brahmi-based script used mainly for writing Marathi (ISO 639-3: mar), an Indo-Aryan language spoken in western and central India, predominantly in the state of Maharashtra.

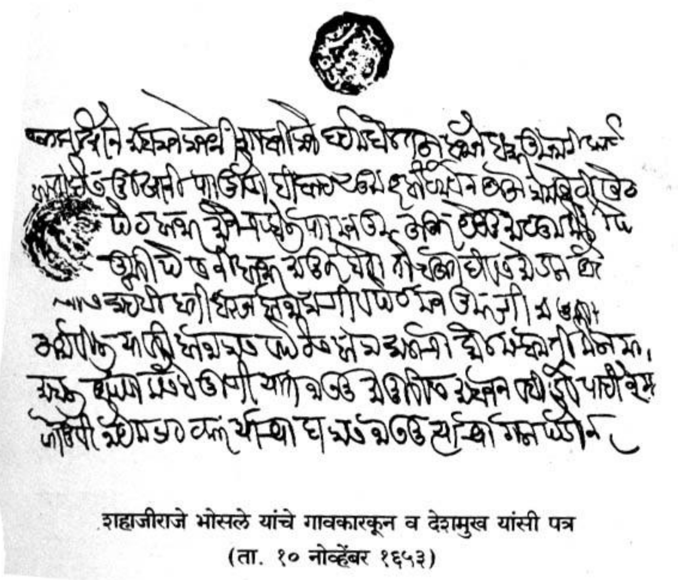


Figure 1. A letter written by Shahaji Raje Bhonsle in the Bahamani style of Modi.

Modi was also used for writing various other regional languages such as Hindi, Gujarati, Kannada, Konkani, Persian, Tamil, and Telugu. According to an old legend, the Modi script was brought to India from Sri Lanka by Hemadri Pandit, known also as Hemad Pant, who was the chief minister of Ramachandra (1271–1309), the last king of the Yadava dynasty. Another tradition credits the creation of the script to Babaji Avaji, the secretary of state to the Maratha king Shivaji Raje Bhonsle (1642–1680), also known as Chhatrapati Shivaji Maharaj. While the veracity of such accounts are difficult to ascertain, it is clear that Modi derives from the Nagari family of scripts and is a modification of the Nagari model intended for continuous writing. More historically, Modi emerged as an administrative writing system in the 16th century before the rise of the Maratha dynasties. It was adopted by the Marathas as an official script beginning in the 17th century and it was used in such a capacity in Maharashtra until the middle of the 20th century. In the 1950s the use of Modi was formally discontinued and the Devanagari script, known as ‘Balbodh’, was promoted as the standard writing system for Marathi. A revival of Modi has occurred over the past decade and its user community continues to grow. Modi users have developed support for using the script on computers, mainly in the form of digitized fonts. But given the lack of a character-encoding standard for the script, usage of these fonts are based upon legacy encodings or are mapped to Unicode blocks such as Devanagari. Electronic materials are produced as images or in Portable Document Format (PDF).

2.1 Writing System Details of Modi Script:

Structure

The general structure (phonetic order, *matra* reordering, use of *virama,* etc.) of Modi is similar to that of Devanagari. Several consonant-vowel combinations are written as ligatures. Consonant clusters are represented as conjuncts. Some consonants have special behaviors when they occur in certain environments

Figure 2. Writing style in Modi

Styles

There are several styles of Modi. The earliest is the 'proto-Modi' of the 12th century, known as Adyakalin. A distinct Modi form emerges during the 13th century and is known as Yadavakalin. The next stage of development is the Bahamanikalin of the 14th–16th century, followed by the Shivakalin of the 17th century. The well-known Chitnisi form develops during this period. In the 18th century, various Modi styles began to proliferate. This era is known as Peshvekalin, which lasted until 1818. The distinct styles of Modi used during this period are known as Chitnisi, Bilavalkari, Mahadevapanti, and Ranadi. The final stage of Modi is associated with English rule and is called Anglakalin. These forms

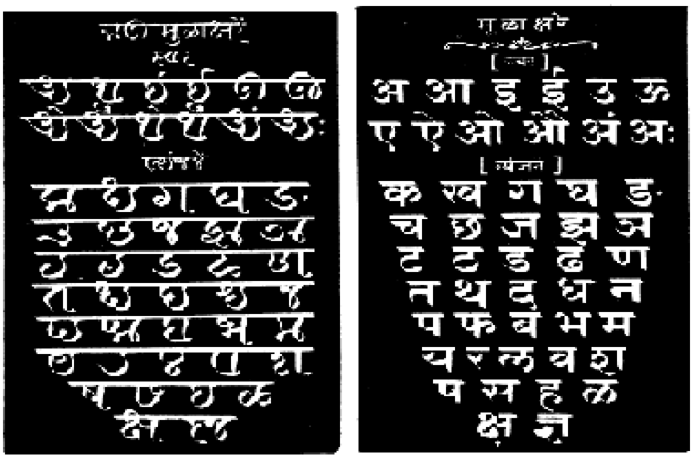


Figure 3: Comparison between Marathi and Modi characters.

were used from 1818 until 1952. Four of the most well-known historical forms are Bahamani, Chitnisi, Peshve and Anglakal. Another style of Modi was used in the primary school books produced during the 19th and 20th centuries. This form was not written in the typical cursive style, a feature that was consciously avoided in order to ensure legibility.

III. SYSTEM DEVELOPMENT

Character Recognition system is divided in two parts. One is training mode and other is recognition mode. During training mode database is prepared and store the results. And during recognition mode sample character is compared with stored patterns in database and computes the result. Major stages in the OCR are I. Pre-processing, 2. Feature Extraction, 3.Recognition, 4. Post processing. [4]. In this research project image pre-processing, features extraction and classification algoritams have been explored to design high performance OCR software for Indian Language Nitodi script. The best performance obtained with fuzzy logic and extracting features using affine moment invariant method.

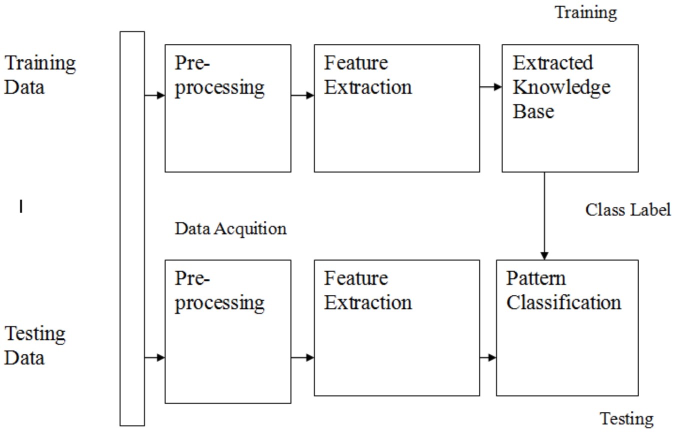


Figure 4. System Development of OCR for Modi script

3.1 Image Preprocessing

Preprocessing is done prior to the application of segmentation and feature extraction algorithms. The raw data is subjected to a number of preliminary processing steps to make it usable in the descriptive stages of character analysis. Preprocessing aims to produce clean document images that are easy for the OCR systems to operate accurately. The main objectives of preprocessing are;

- Noise reduction
- Normalization of the data
- Compression in the amount of information to be retained.

Data in a paper document are usually captured by optical scanning and stored in a file of picture elements, called pixels. These pixels may have values: OFF (0) or ON (1) for binary images, 0— 255 for gray-scale images, and 3 channels of 0-255 color values for color images. This collected raw data must be further analyzed to get useful information. Such processing includes the following:

3.2 Thresholding:

In order to reduce storage requirements and to increase processing speed, it is often desirable to represent grey scale or color images as binary images by picking some threshold value for everything above that value is set to 1 and everything below is set to 0 Two categories of thresholding exist: Global and Adaptive. Global thresholding picks one threshold value for the entire document image, often based on an estimation of the back ground level from the intensity histogram of the image. Adaptive thresholding is a method used for images in which different regions of the image may require different threshold values.

3.3 Noise reduction

The noise, introduced by the optical scanning device or the writing instrument, causes disconnected line segments, bumps and gaps in lines, filled loops etc. The distortion including local variations, rounding of corners, dilation and erosion, is also a problem. Prior to the character recognition, it is necessary to eliminate these imperfections. Filtering: The aim of filtering is to remove noise and diminish spurious points usually introduced by uneve: writing surface and or poor sampling rate of the image acquisition device Various spatial an frequency domain filters can be designed for this purpose. The basic idea is to convolve a pr(defined mask with the image to assign a value to a pixel as a function of the gray values of it neighboring pixels One example is to use linear spatial mask where the intensities x(i, j) 01 the input image is transformed to the output image by

$$v(i,j) = \sum_k \sum_i a_{kl}\, x(i-k, j-l)$$

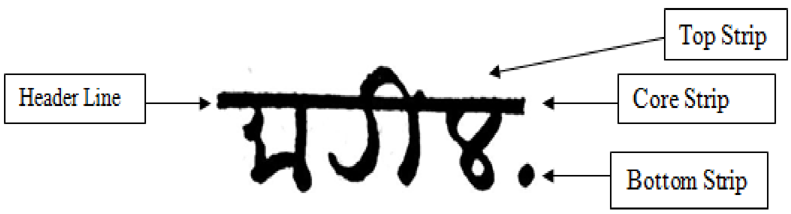
where a_{kl} is the weight of the gray levels of pixels of the mask at location (k, 1)Filters can be designed for smoothing sharpening thresholding and contrast adjustment purposes.

3.3.1 Normalization:
It is used to adjust the character size to a certain standard. Methods of character recognition may apply both horizontal and vertical size normalizations. Each segmented character is normalized to fit within suitable matrix like 50x50 or 40x40 so that all characters have same data size. [7,8,9]

3.4 Segmentation
The preprocessing stage yields a "clean" document in the sense that maximal shape information with maximal compression and minimal noise on normalized image is obtained. The next stage is segmenting the document into its sub components and extracting the relevant features to feed to the training and recognition stages. Segmentation is an important stage, because the extent one can reach in separation of words, lines or characters directly affects the recognition rate of the script. There are two types of segmentation.

3.4.1 External Segmentation:
External segmentation is the isolation of various writing units, such as paragraphs, sentences or words, prior to the recognition. External segmentation decomposes the page layout into its logical parts. It provides savings of computation for document analysis. Page layout analysis is accomplished in two stages. The first stage is the structural analysis, which is concerned with the segmentation of the image into blocks of document components (paragraph, rows, words, etc). The second is the functional analysis, which uses location, size and various layout rules to label the functional content of document components (title, abstract, etc). Structural analysis may be performed in top-down bottom-up or in some combination of the two approaches.

3.4.2 Internal Segmentation:
This is an operation that seeks to decompose an image of a sequence of characters into sub images of individual symbols. It is one the most important process that decides the success of character recognition technique. It is used to decompose an image of a sequence of characters into sub images of individual symbols by segmenting lines and words. In this project work the line segmentation is performed first. The image of text may contain any number of lines. Thus, first need to separate the lines from the documents and then proceed further. This is referring to as line segmentation. To perform line segmentation, take horizontal projection for every horizontal pixel row starting from the top of document. The lines are separated for a row with no black pixels. That means, HP (k) = 0 where k is the row number where white space is found. This row acts as a separation between two lines. Then word segmentation is performed in this word boundaries are detected by looking for the vertical gaps in the segmented line, and checking them to identify the beginning and end of words. Then character segmentation is performed, after detection of reference line it is removed from the word to separate out the characters again by looking for vertical gaps in the segmented word, and checking them to identify the beginning and end of character. Various vowel modifiers can be separated for structural feature extractions. Composition Of Characters And Symbols For Writing Words is As Follows: A horizontal line is drawn on top of all characters of a word that is referred to as the header line or shirorekha. It is convenient to visualize a mali word in terms of three strips: a core strip, a top strip and a bottom strip. The core and top strips are separated by the header line. Figure 4.3 shows the image of a word that contains nine characters, a top modifier, and one lower modifier. The three strips and the header line have been marked. No corresponding feature separates the lower strip from the core strip. The top strip has top modifiers and the bottom strip has lower modifiers whereas the core strip has the characters and core modifier. If no consonant of a word has a top modifier, the top strip will be empty. Similarly, if no character of a word has a lower modifier, the bottom strip will be empty. It is possible that either of the bottom or top strips may be present or both may be present.



Before describe the preliminary segmentation of words, let us first define the following two operators:

1. Definition 1 : Vertical Projection: For a binary image of size H * W where H is the height of the Image and W is the width of the image, the vertical projection has been defined as V P(k), k = 1, 2,……..,W This operation counts the total number of black pixels in each vertical column.
2. Definition 2: Horizontal Projection: For a binary image of size H * W where H is the height of the image and W is the width of the image, the horizontal projection is defined as H P (j), j = 1, 2,…..,H This operation counts the total number of black pixels in each horizontal row.

The preliminary segmentation consists of the following five steps:
Step I: Locate the text lines A text line is separated from the previous and following text lines by white space. The line segmentation is based on horizontal histograms of the document. Those rows, for which HP[j] is zero; j = 1, 2,,H; serve as delimiters between successive text lines.

Step II: Locate the words The segmentation of the text line into words is based on the vertical projection of the text line. A vertical histogram of the text line is made and white space are used as word delimiter.

Step III: Locate the header line After extracting the sub images corresponding to words for a text line, we locate the position of the header line of each word. Coordinates of the **top-left** corner are (0,0) and bottom-right corner are (W, H) where H is the height and W is the width of the word image box. We compute the horizontal projection of the word image box. The row containing maximum number of blackpixels is considered to be the header line. Let this position be denoted by hLinePos. Figure (a) shows image of a word and figure (b) shows its horizontal projection. The row that corresponds to the header line has been marked as hLinePos.

Step IV: Separate character/symbol boxes of the image below the header line: To do this, make vertical projection of the image starting from the hLinePos to the bottom row of the word image box. The columns that have no black pixels are treated as boundaries for extracting image boxes corresponding to characters. Figure (c) shows the vertical projection. The columns corresponding to white space between successive characters have been marked. The extracted sub images have been shown in figure(d).

Step V: Separate symbols of the top strip To do this, compute the vertical projection of the image starting from the top row of the image to the header h Line Position. The columns that have no black pixels are used as delimiters for extracting top modifier symbol boxes.[9,12]

IV. FEATURE EXTRACTION

A feature point is a point of human interest in an image, a place where something happens. It could be an intersection between two lines, any type of distance between pixels or it could be a corner, or it could be just a dot surrounded by space. Such points serve to help define the relationship between different character images. Selecting features according to some criterion amounts projecting it onto a particular view, which usually greatly simplifies the data. A good property of a feature is comparability. That is feature extraction should enable comparison between characters by simple comparisons on the features. During or after the segmentation procedure the feature set, which is used in the training and recognition stage, is extracted. Feature sets play one of the most important roles in a recognition system. Feature extraction and selection can be defined as extracting the most representative information from the raw data, which minimizes the within class pattern variability while enhancing the between class pattern variability. For this purpose, a set of features are extracted for each class that helps distinguish it from other classes, while remaining invariant to characteristic differences within the class. Following feature extraction methods are implemented.

V. AFFINE MOMENTS INVARIANTS (AMI)

Affine moment invariants (AMIs) I] are important tools in object recognition problems. These techniques are commonly divided into two main categories according to how they make use of the image function. In so called local approaches the objects are segmented to smaller elements and invariants are computed separately for each of them. The second category consists of the global approaches [6], where the features are computed directly from the whole image intensity function. The AMIs were derived by means of the theory of Algebraic invariants. We used the AMIs are invariant under general affine transformation, therefore

$u = a_o+a_1x+a_2y$ (5.1)

$v = b_o+b_1x+b_2Y$ (5.2)

where (x,y) and (u,v) are coordinates in the image plan before and after the transformation , respectively. Four simplest AMIs that we have used for character recognition are listed below: [11,14]

$I_1 = 1/\mu^4{}_{00}(\mu_{20}\mu_{02} - \mu^2{}_{11})$

$I_2 = 1/\mu^{10}{}_{00}(\mu^2{}_{30}\mu^2{}_{03} - 6\mu_{30}\mu_{21}\mu_{12}\mu_{03} + 4\mu_{30}\mu^3{}_{12} + 4\mu_{03}\mu^3{}_{21} - 3\mu^2{}_{21}\mu^2{}_{12})$

$I_3 = 1/\mu^7{}_{00}(\mu_{20}(\mu_{21}\mu_{03} - \mu^2{}_{12}) - \mu_{11}(\mu_{30}\mu_{03} - \mu_{21}\mu_{12}) + \mu_{02}(\mu_{30}\mu_{12} - \mu^2{}_{21}))$

$I_4 = 1/\mu^{11}{}_{00}(\mu^3{}_{20}\mu^2{}_{03} - 6\mu^2{}_{20}\mu_{11}\mu_{12}\mu_{03} + 9\mu^2{}_{20}\mu_{02}\mu_{12} + 12\mu_{20}\mu^2{}_{11}\mu_{21}\mu_{03} + 6\mu_{20}\mu_{11}\mu_{02}\mu_{30}\mu_{03} - 18\mu_{20}\mu_{11}\mu_{02}\mu_{21}\mu_{12} - 8\mu^3{}_{11}\mu_{30}\mu_{03} - 6\mu^2{}_{20}\mu_{02}\mu_{12} + 9\mu_{20}\mu^2{}_{02}\mu_{21} + 12\mu^2{}_{11}\mu_{02}\mu_{30}\mu_{12} + 6\mu_{11}\mu^2{}_{02}\mu_{30}\mu_{21} + \mu^3{}_{02}\mu^2{}_{30})$

These equations are used to evaluate 4 Affine Invariant Moments (i.e. $I_1 – I_4$) which are used as features. Table 1 shows four affine invariant features of letter 'AA'

Mean of letter AA			
I_1	I_2	I_3	I_4
34.8683	102.9935	69.7804	r 102.6295
35.1458	104.1306	69.9612	105.6915
34.6417	100.5367	67.5049	99.9122
34.9277	103.8941	69.1556	101.6918
Mean of **I1**	Mean of **I2**	Mean of **I3**	Mean of **I4**
34.7337	101.2555	68.1524	1 0 1 . 2 3 6 4
Standard Deviation			
0.4047	3.9156	2.3308	3.4836

Table 1. Affine Invariant features of letter 'AA'

VI. CLASSIFICATION

6.1 Fuzzy logic:

Fuzzy logic plays a major role in the area of character recognizing due to the following reasons:

- Fuzzy logic is a precise logic that deals with the impreciseness which arises due to the lack of knowledge.
- Fuzzy_features_description model human perception of features such as, 'VERY STRAIGHTLINE', 'VERY SMALL CURVE', and so on .therefore, a fuzzy rule base consisting of these linguistic rules would be very flexible.

The mathematical computation required by a fuzzy system are very fundamental (i.e. min, max, addition and multiplication).thus the calculations involved in fuzzy features description are extremely short and simple. Therefore, any fuzzy system could be efficiently executed even with low computational requirements [5].

6.2 Fuzzy Gaussian Membership Function

A database of handwritten vowels will required to extract the feature, which form a Trained database. The reference character dataset has been obtained from training samples. The mean and standard deviation are computed for each type of features. The mean and standard deviation can be computed for each type of features. Therefore there are respective mean and standard deviation corresponding to each value of the characters. The unknown character features available in the template and found the maximum membership value (0-1) of the character with the reference vowels. The template consists of means M, and cr, for each feature and is computed as: [11,13] means Mi, and σi, for each feature and is computed as: [11,13]

$$\text{Mean } M_i = \frac{1}{N_i}\sum I_{i(k)} \qquad (5.3)$$

$$\text{Std-Dev } \sigma_i = \sqrt{\sum (I_{i(k)} - M_i)^2}$$

Where N, is the number of samples in I^{th} class and $I_{i(k)}$ Stands for the k^{th} feature value of reference character in the i^{th} class. In this work, for unknown input character X, the corresponding features will be extracted. The Fuzzy Gaussian Membership Function [15] will be attempted to get the maximum membership values as follows:

$$\mu_{xi} = \exp - \frac{(x_i - M_i)}{2\sigma_i^2} \qquad (5.5)$$

Where Xi is the i^{th} feature of the unknown character. Let Mj(r), $\sigma j^2(r)$ belongs to the r^{th} reference character with r = 0,1, 2,.....,6. We then calculate the average membership value as

$$\mu_{av}(r) = \frac{1}{C}\sum_{j=1}^{C} \exp - \frac{(x_i - M_i)}{2\sigma_i^2} \qquad (5.6)$$

Where X ε r if μ_{av} (r) is the maximum for r = 0, 1, 2,...6 The resultant array is then normalized in order to enhance the success rate by the following expressions.

$$d_i(t) = \frac{1}{N}\sum_1^N d_1 \qquad (5.7)$$

Where N is the number of classes in the group, d_i is i^{th} feature of class d.

$$\text{Normalization } n = \left(\frac{d_i}{d_i(t)}\right) * 100 \qquad (5.8)$$

VII. RECOGNITION

The objective of recognition is to recognize the character taken from the test set Any new characters that is to be recognized is preprocessed first. Feature extracted from this character are used to find mean and standard deviation of these characters. The unknown character features available in the each group and found the maximum membership value(0-1) of the character with the reference character in the corresponding group. This mean and standard deviation is given as input to the Gaussian Membership Function and we get the output result as the recognized images.

VIII. CONCLUSION

It is common sense that if an OCR system can achieve an excellent recognition performance, the following two aspects must have played an important role: feature extraction and classification. In this, our research focuses on: feature extraction using AMIs technique in order to increase the recognition accuracy and reliability. A database of characters has been required to extract the features, which form a template (Trained Database). In character experiments, we have used four AMIs. The experiments have confirmed the following claims:

- If the deformation between the templates and the unknown characters is approximately affine, the classification based on AMIs yields high reliable results.
- If the character deformation is more complex (non-linear), some error may occur in the classification. The number of error can be reduced by composing of the training set from letters of various fonts.

To improve the performance rate the hybrid features based on image structure and statistic can be added.
The main advantage of our approach is that the features are invariants under translation, scale, rotation and reflection. It overcomes the problem in varieties in handwriting. Some time the handwritten character are tilt, small or large in shape. The AMIs features are near about same for these variations. But some character features create complexity. We found the solution in form of box method and also other feature extraction methods for solving the problem as they help in enhancement of success rate in spite of great variation in character due to different styles of handwriting which is our future scope of research work.

REFERENCES

[1]. Anshuman Pandey, Proposal to Encode the Modi Script in ISO/IEC 10646, ISO/IEC JTC1/SC2/WG2 N4034, L2/11-212R2 2011-11-05

[2]. Sadanand A. Kulkarni, Prashant L. Borde, Ramesh R. Manza, Pravin L. Yannawar, Offline Handwritten Modi character recognition using HU, Zernike Moments and Zoning 2013

[3]. D. N. Besekar, R. J. Ramteke, International Journal of Computer Applications, vol. 64, no. 3, February 2013. "Study for Theoretical Analysis of Handwritten MODI Script — A Recognition Perspective.

[4]. Ovind Trier, Anil Jain and Torfinn taxt,"A feature extraction methods for character recognition-A survey", pattern recognition, vol 29, no-4, pp 641-662, 1996.

[5]. U.Pal and B.B. Chaudhuri," An improved document skew angle estimation techniques" Pattern Recognition Letters 17:899-904, 1996.

[6]. B.B. Chaudhruri and U.Pal, "A complete printed OCR", Pattern Recognition,(5):531-549, 1998.

[7]. N.Arica, Vural, "An overview of Character recognition focused on offline Handwriting", IEEE trans on system , man, cybernatics-Partc , vol 31,N0.2(2001).

[8]. R.J. Ramteke, S.C. Mehrotra "Feature extraction based on Invariants Moment for handwritten Recognition", Proc. of 2nd IEEE Int. Conf. On Cybernetics Intelligent System(CIS2006),Bangkok, June 2006.

[9]. T.V.Aswin and P S Sastry, " A font and size-independent OCR system for printed Kannada documents using support vector machines", Sadhana Vol.27.part I,pp.35-58,Februvary 2002.