International e-Conference on New Horizons And Multidisciplinary Applications In Science And Technology
In Association withInternational Journal of Scientific Research in Science and Technology
Volume 9 | Issue 6 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X (www.ijsrst.com)

Recognition of Devanagari CAPTCHA Code Using Novel Feature Extraction Methods and PNN Classifier

P.S. Bodkhe[1], Dr. P.E. Ajmire[2]
[1]Associate Professor, Department of Computer Science, G.S. College, Khamgaon, Maharashtra, India
[2]Head & Professor, Department of Computer Science, G.S. College, Khamgaon, Maharashtra, India

ABSTRACT

The CAPTCHA is a test deployed on most of the websites to safeguard its services from illegal users.. It stands for Completely Automated Public Turing Test to Tell Computers and Humans Apart. The CAPTCHA test is used to classify between authentic user and unauthentic user. The authentic user is always a human and unauthentic user is an automated program called bot. The CAPTCHA code should be such that it can be easily recognized by a human, but at the same it must be difficult for illegal bots. Thus the CAPTCHA is like a gatekeeper who checks the legacy of the user. Some of the Indian government/non-government websites have started publishing the information and offering various online services in regional languages. Most of the regional languages are derived from Devanagari. But, the CAPTCHA employed during log-in time, is still in English script, which reduces the accessibility of websites. The native users are always comfortable if the CAPTCHA test is offered in their native language. Therefore, in order to simplify the task, this paper emphasize on employing a Devanagari CAPTCHA in place of English CAPTCHA. The recognition of Devanagari CAPTCHA code is always a challenging task. The work in this paper, proposed a system that use some novel feature extraction methods like various shape descriptors and a statistical method called moment invariant. In order to recognize the characters in Devanagari CAPTCHA code, the extracted features are classified by using Probabilistic Neural Network (PNN) classifier. The achieved recognition accuracy of the Devanagari characters by using PNN is better as compared to other techniques.

Keywords: PNN, Shape Descriptors, Devanagari, script.

I. INTRODUCTION

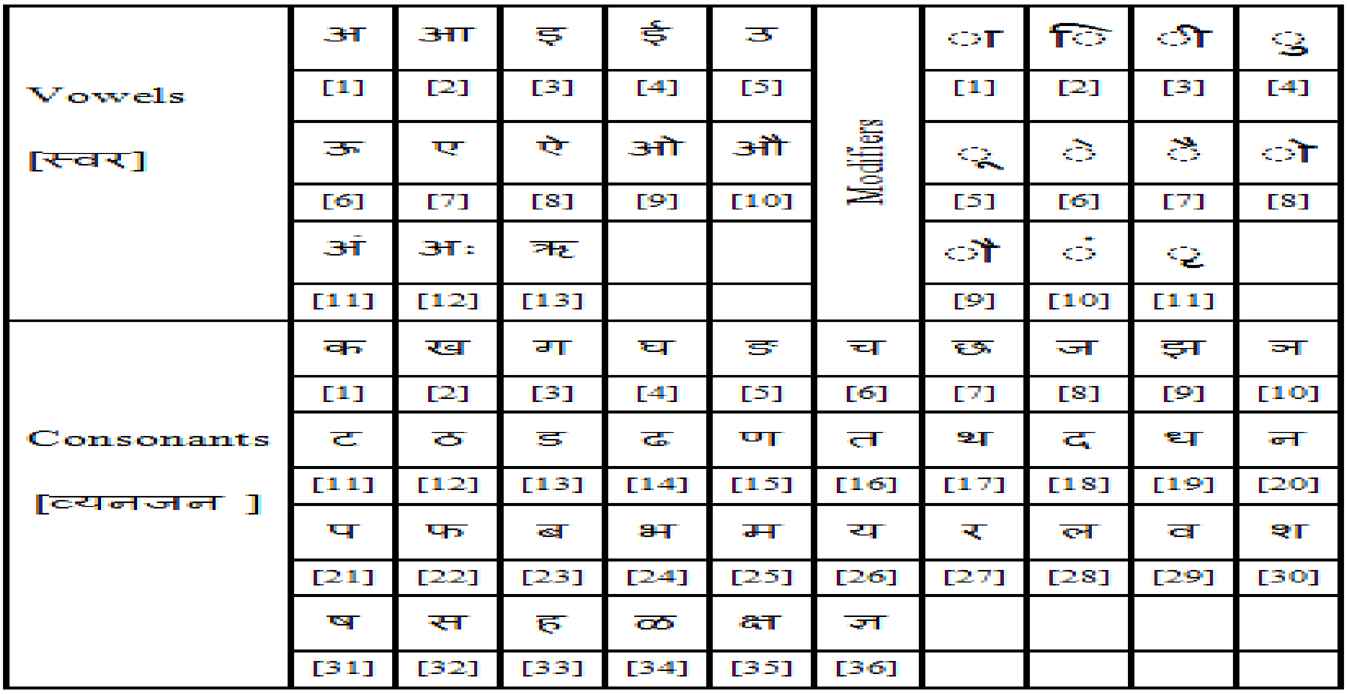
All most every important website has incorporated a security test called CAPTCHA to authenticate its users. This test is used during login time. CAPTCHA is a challenge-response test to authenticate that the user is a human and it also ensures that the response is not generated by any automated computer bots [1]. CAPTCHAs are actually a type of Human Interaction Proofs (HIP) [1]. This process involves one computer asking a user to complete a test. The CAPTCHA test normally consist of alphabetic characters, numerals, images or audio that any user entering a correct response is accepted as a human and the user failing to enter the correct response is determined as a robot. The permission to access the website is denied if a user fails to pass the test.

The purpose is to create a test that the human can pass it easily but not the computer bots [1]. This paper proposed the process of recognizing characters of Devanagari CAPCHA code. It is based on various feature extraction methods that includes various shape descriptors: Convex Area, Filled area, Euler Number, Eccentricity, EquivDiameter, Centroid, Bounding Box and invariant moments. In order to recognize the Devanagari CAPTCHA code characters, Probabilistic Neural Network (PNN) classifier is used to classify the extracted features. The proposed method has been implemented on Devanagari characters. These characters are used as an input by applying font effects such as Bold, Italic, rotated left and right with specified degrees, and adding background noise.

II. DEVANAGARI SCRIPT

The Devanagari script is one of the ancient and widely used scripts of India [2]. It is adopted by more than 120 Indo-Aryan languages which include Sanskrit, Hindi, Marathi, Pali, Awadhi, Konkani, Bodo, Bhojpuri, Newari, Maithili and Nepali languages [3]. Beside this, it also used as a supportive script for other Indian languages such as Punjabi, Sindhi and Kashmiri making it one of the commonly used and adopted writing systems in the world [3]. It is the fourth mostly adopted script in the world [4].

Devanagari script has 49 primary characters which includes 13 vowels, 36 consonants and 11 modifiers [5], as shown in following figure 1.



Vowels [स्वर]	अ	आ	इ	ई	उ	Modifiers	◌ा	ि◌	◌ी	◌ु
	[1]	[2]	[3]	[4]	[5]		[1]	[2]	[3]	[4]
	ऊ	ए	ऐ	ओ	औ		◌ू	◌े	◌ै	◌ो
	[6]	[7]	[8]	[9]	[10]		[5]	[6]	[7]	[8]
	अं	अः	ऋ				◌ौ	◌ं	◌ृ	
	[11]	[12]	[13]				[9]	[10]	[11]	
Consonants [व्यनजन]	क	ख	ग	घ	ङ	च	छ	ज	झ	ञ
	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
	ट	ठ	ड	ढ	ण	त	थ	द	ध	न
	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]	[19]	[20]
	प	फ	ब	भ	म	य	र	ल	व	श
	[21]	[22]	[23]	[24]	[25]	[26]	[27]	[28]	[29]	[30]
	ष	स	ह	ळ	क्ष	ज्ञ				
	[31]	[32]	[33]	[34]	[35]	[36]				

Fig.1: Devanagari character set

From the Sanskrit language, "Deva" means God and "Nagari" means city/lipi/script [2]. Therefore, Devanagari is a "Script of God" or “Language of God”. Devanagari script is commonly used to write languages such as Sanskrit, Hindi, Marathi, Konkani, Maithili, Bhojpuri, Rajasthani, Haryanvi and Nepali.

There are officially 22 languages and 13 major scripts in India. Devanagari is one of the widely used scripts [3]. It is used to write Hindi, which is the third mostly spoken language in the world [2]. There are nine major Indian states whose official language is Hindi. These states are: Madhya Pradesh, Rajasthan, Himachal Pradesh, Madhya Pradesh, Haryana, Uttarakhand, Chhattisgarh, Zarkhand and Delhi [3].

III. IMPORTANCE OF DEVANAGARI CAPTCHA

Devanagari script-based CAPTCHA is used by a large number of Indian languages including Hindi, which is the third most spoken language in the world [6]. Many other official languages like Marathi, Gujrathi, Bhojpuri and Rajasthani are also used on large scale.

Most of the Indian Government websites provide its contents in Devanagari script based languages like Hindi, Marathi, Haryanvi, and Gujrati [7]. But to secured its contents from any misused by an unauthorized computer bots, the CAPTCHA test is generated in English language. Many regional users, who know only their native language, face difficulties in passing CAPTCHA test. Thus, to improve the accessibility of the websites, the CAPTCHA test needs to be constructed in their own languages, which are mostly originated from Devanagari script.

The proposed system offers the CAPTCHA test in Devanagari script. The aim is to increase the usability and security of the Indic websites by incorporating strong CAPTCHA test in the languages which are originated from Devanagari script [8].

IV. DATABASE DESIGN

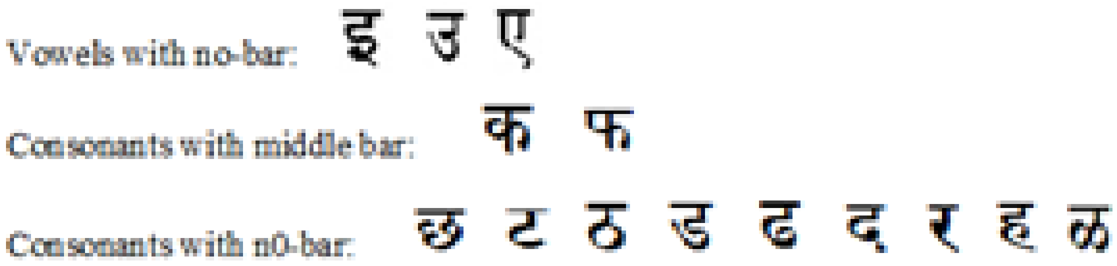
In order to accomplish the task of Devanagari character recognition, the very first step is to prepare a database of its characters. At present, no such a dataset is available on Internet for Devanagari characters. Hence, the database is prepared using different Devanagari fonts, considering only commonly used basic consonants, limited vowels and numerals.

The specific dataset of limited characters is prepared in order to improve the usability and security of Devanagari CAPTCHA script. The purpose is to make the character recognition task easier for the user while going through CAPTCHA test and should be difficult for the computer bots.

There are 34 consonants and 13 vowels in Devanagari script. Devanagari consonants and vowels are categorized according their structural properties. Therefore, the CAPTCHA code containing Devanagari letters can be divided into three groups based on the presence and position of a vertical bar, namely:

1. End bar characters,
2. Middle bar characters and
3. No-bar Characters

Accordingly, the 23 Devanagari consonants and 1 vowel that terminates woth end-bar, the 9 Devanagari consonants with no-bar, 2 with middle-bar and 3 vowels with no-bar (Figure 3) along with 10 numeric characters, from 5 different fonts of Devanagari characters are considered to prepare a dataset. The database will consist of 48 Devanagari characters.

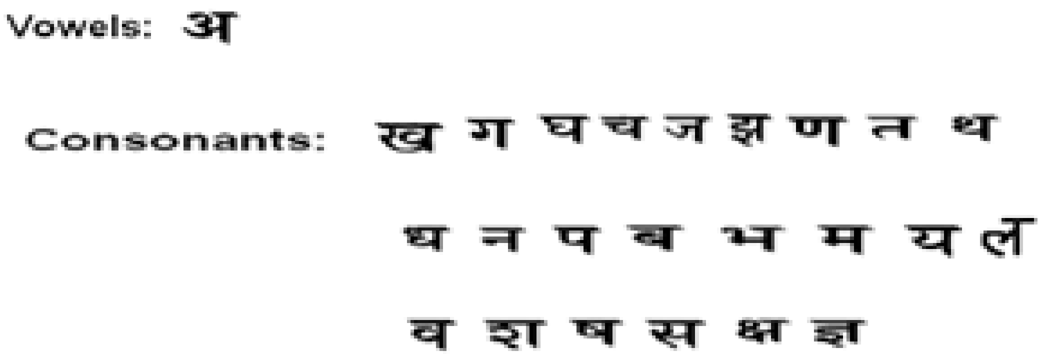


Vowels with no-bar: इ उ ए

Consonants with middle bar: क फ

Consonants with n0-bar: छ ट ठ ड ढ द र ह ळ

Fig.2: Devanagari characters with middle-bar and no-bar



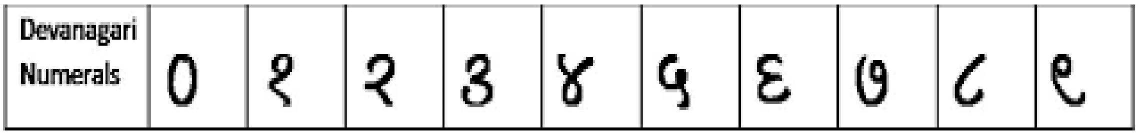
Vowels: अ

Consonants: ख ग घ च ज झ ण त थ

ध न प ब भ म य ल

व श ष स क्ष ज्ञ

Fig.3: Devanagari characterswith end-bar.



Devanagari Numerals	०	१	२	३	४	५	६	७	८	९

Fig.4: Devanagari numeric characters(digits).

The algorithm used for designing database is given below:

Algorithm: To Create Datasheet

Step1. The selected character is typed in MS-Word.

Step2: The character is stored in Excel sheet.

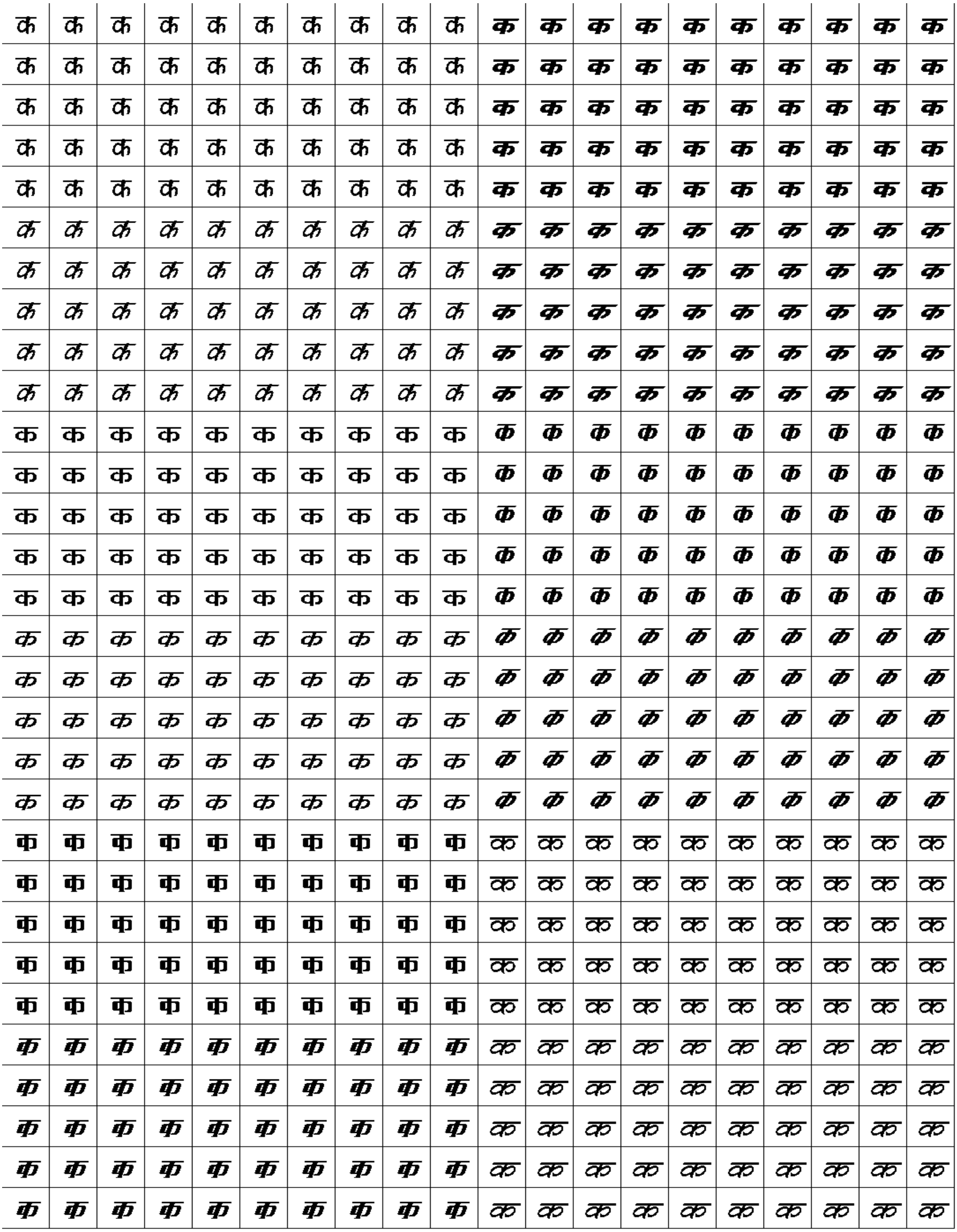
Step3: Preprocessed the character.

Step4: Resize the character.

Step5: The steps 1 to 4 are performed for 34 characters.

Algorithm 5.1: Generation of Datasheet.

As an example, a sample of datasheet created for Devanagari letter ‘ क ‘ is shown in figure 5.



क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क
क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क
क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क
क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क
क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क
क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क
क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क
क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क
क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क
क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क
क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क
क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क
क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क
क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क
क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क
क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क
क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क
क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क
क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क
क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क
क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क
क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क
क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क
क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क
क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क
क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क
क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क
क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क
क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क
क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क	क

Fig.5: Sample Data sheet of letter ' क '

The above algorithm will result in creation of 48 datasheets for 48 Devanagari characters. All the datasheets created are stored in one directory. Each sheet of characters is then segmented into separate characters with the help of following algorithm:

Algorithm: To convert datasheet into a database.

```
Step1:  Read Datasheet file
Step2:  Convert to gray
Step3:  Convert gray to binary
Step4:  Take compliment of this data file
Step5:  Crop each character
Step6:  Resize each character to 40 x 40 size
Step7:  Store each character in same directory.
```

Algorithm 5.2: Conversion of Datasheet into a database.

The above algorithm results in creation of 250 images of each character which are then stored in a separate file. The database created, thus comprises of 12000 (48X250) images. In this way the individual character is available in the form of image in database. Some sample characters from the database are shown in figure 6.

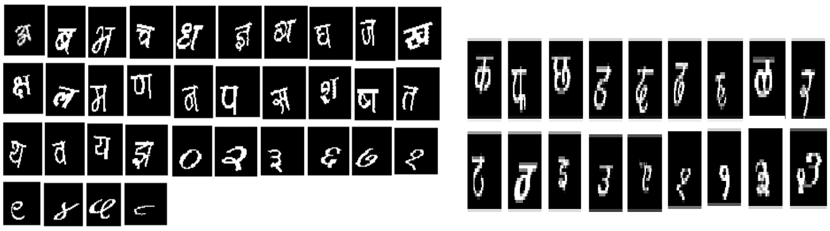


Fig. 6: Sample of character images taken from dataset.

V. FEATURE EXTRACTION METHODS

Feature extraction is commonly used to obtain the characteristics and important features of the given character image. Feature extraction is one of the important stages in pattern recognition. If you used right feature extraction technique, then it leads to achieve higher character recognition rate. In order to extract features of characters, the regional descriptors like Area, Solidity, BoundingBox, Centroid, ConvexArea, Eccentricity, EquivDiameter, EulerNumber, Extent, FilledArea, MajorAxisLength, MinorAxisLength and statistical method like Moment Invariants are used. Seven moment invariants are evaluated for each character along with all these descriptors. The various shape descriptors and moment invariant methods used to extract features, are briefly described.

- **Area:** Area is given by the number of pixels of which the region consists. The real area of each pixel may be taken into consideration to get the real size of a region. If an image is represented as a rectangular raster, simple counting of region pixels will provide its area [9].
- **Bounding Box:** The bounding box or bounding rectangle of an object is a rectangle which circumscribes the object. The dimensions of the bounding box are those of the major and minor axes. The area of the bounding box is [9]:
  - Area of bounding box = (Major axis length x (Minor axis length)
- **Convex area:** The convex area of an object is the area of the convex hull that encloses the object [9].

- **Centroid:** It is the measured as a center of mass of the region. The first element of Centroid is the horizontal coordinates (or x-coordinate) of the center of mass, and the second element is the vertical coordinate (or y-coordinate).
- **EquivDiameter:** It is the diameter of a circle with the same area as the region and it is computed as sqrt (**4xArea/pi**) [9].
- **Extent:** It considers the proportion of pixels in the bounding box that are also in the region and Centroid is computed as Area divided by the area of the bounding box [9].
- **FilledArea:** FilleArea is the number of 'on' pixels in FilledImage. Filled image is the binary image of the same size as the bounding box of the region. The 'on' pixels are corresponding to the region with all holes filled [9].
- **Euler's number:** It describes a simple topologically invariant property of the object. S is the number of contiguous parts of an object and N is the number of holes in the object (an object can consist of more than one region) [9]. $\vartheta = S - N$
- **Eccentricity:** The eccentricity is the ratio of the distance between the foci of the ellipse and its major axis length. The value is between 0 and 1 being degenerate cases (an ellipse whose eccentricity is 1 is a line segment, while an ellipse with an eccentricity of 1 is a circle [9].
- **Perimeter:** Perimeter is number of pixels in the boundary of the shape. How closely-packed the shape is: perimeter^2/area. The most compact shape is a circle (4π). All other shapes have compactness larger than 4π. If x_1,…, xN is a boundary list, the perimeter is given by: –

  Perimeter = $\sum_{j=1}^{N-1}$ **di** = $\sum_{j=1}^{N-1}$ **| Xi − Xi + 1 |**

  The distances di are equal to 1 for 4-connected boundaries and 1 or for 8-connected boundaries [9].
- **Major Axis Length:** The major axis endpoints (x1, y1) and (x2, y2) are found by computing the pixel distance between every combination of border pixels in the object boundary and finding the pair with the maximum length. The major-axis length of an object is the pixel distance between the major-axis endpoints and is given by the relation [9]:

$$\text{Major-axis length} = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}$$

- **Minor Axis Length:** The minor axis endpoints (x1, y1) and (x2, y2) are found by computing the pixel distance between the two border pixel endpoints. The minor-axis length of an object is the pixel distance between the minor-axis endpoints and is given by the relation [9]:

$$\text{Minor-axis length} = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}$$

- **Solidity:** Solidity measures the density of an object. • A measure of solidity can be obtained as the ratio of the area of an object to the area of a convex hull of the object: **Solidity = Area / Convex Area** [9].
- **Moment Invariants:** The way to describe the shape that uses statistical properties is called moments. Region moment representations interpret a normalized gray level image function as a probability density of a 2D random variable. Properties of this random variable can be described using statistical characteristics - moments. Assuming that non-zero pixel values represent regions, moments can be used for binary or gray level region description. A 2-D moment of order (s + t) of a digital image f(x, y) of size m x n is dependent on scaling, translation, rotation, and is given by [9]:

$$m_{st} = \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} x^s y^t f(x, y),$$

Where s = 0, 1, 2…and t=0, 1, 2…are integers, x, y are the region point co-ordinates (pixel co-ordinates in digitized images).

The corresponding central moment of order (s + t) is define as follows:

$$H_{st} = \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} (x - x?)^s (y - y?)^t f(x, y)$$

Where s =0, 1, 2… and t=0, 1, 2, …., $\bar{x} = m_{10} / m_{00}$, $\bar{y} = m_{01} / m_{00}$

The moments that are computed with their centroid being about the origin are called central moments, denoted by $\mu_{st.}$ The normalized central moment of order (s + t) is define as follow.

$\eta_{st} = \mu^{\gamma},$ where $\gamma = (s + t) / 2 + 1$

The set of seven moment invariants are derived from these equations, as follows: The invariant moment has 7 moments (ϕ) and they are defined using the normalized central moment, such as:

$\Phi 1 = \eta_{20} + \eta_{02}$

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

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

$\Phi 4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$

$\Phi 5 = (\eta_{30} - 3\eta_{12})\,(\eta_{30} + \eta_{12})\,[\,(\eta_{30} + \eta_{12})^2 - 3\,(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})\,(\eta_{21} + \eta_{03})3\,(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$

$\Phi 6 = (\eta_{20} - \eta_{02})\,[(\eta_{30} + \eta_{12})2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}\,(\eta_{30} + \eta_{12})\,(\eta_{21} + \eta_{03})$

$\Phi 7 = (3\eta_{21} - \eta_{03})\,(\eta_{30} + \eta_{12})\,[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})\,(\eta_{21} + \eta_{03})]\,(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$

The above 7 moments are treated as 7 features of the invariant moment.

These feature extraction methods used to extract the characteristics of each character from the 250 character images and with the help of feature selection algorithm, it generates the feature vector containing 14 features for each character image from dataset of 12000 images.

From the database, 250 samples of each character are used for feature extraction. In this way, the feature is a matrix of 250 x 14 features for each character. To improve the accuracy of extracted features, the value with 4 digits precision is used. As an example, the sample of '14' extracted features of Devanagari character 'क'are given in table 1. The algorithm 5.3 is used for feature extraction of these features.

```
Algorithm: Feature Extraction & Selection
 Step1:  Read the data image from database
 Step2:  Extract feature of image
 Step3:  Store these features
 Step4:  Use feature selection to form features vector
Step5:   Repeat the procedure for each character in database
```

Algorithm 5.3: Feature Extraction & Selection

Table 1: The first 50 sample Features (out of 250) of character 'क'

Features --->

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
क	1.12 25	5.22 61	9.59 89	9.04 27	19.01 68	12.15 69	18.66 85	1.51 59	12.69 05	16.11 78	9.89 56	23.23 91	16.59 93	23. 662 1
क	0.66 33	7.98 60	3.33 70	3.77 14	7.983 3	10.00 82	9.139 8	1.46 06	9.035 0	13.19 55	8.92 26	20.22 71	13.80 64	25. 012 2
कृ	0.66 33	7.98 60	3.33 70	3.77 14	7.983 3	10.00 82	9.139 8	1.46 06	9.035 0	13.19 55	8.92 26	20.22 71	13.80 64	25. 012 2
क	0.66 33	7.98 60	3.33 70	3.77 14	7.983 3	10.00 82	9.139 8	1.46 06	9.035 0	13.19 55	8.92 26	20.22 71	13.80 64	25. 012 2
क	0.66 33	7.98 60	3.33 70	3.77 14	7.983 3	10.00 82	9.139 8	1.46 06	9.035 0	13.19 55	8.92 26	20.22 71	13.80 64	25. 012 2
क	0.65 43	7.93 49	3.31 00	3.75 38	7.950 8	10.18 89	8.956 0	1.46 21	9.042 9	13.20 59	8.93 87	20.25 62	13.82 63	25. 160 5
क	0.66 33	7.98 60	3.33 70	3.77 14	7.983 3	10.00 82	9.139 8	1.46 06	9.035 0	13.19 55	8.92 26	20.22 71	13.80 64	25. 012 2
क	0.66 33	7.98 60	3.33 70	3.77 14	7.983 3	10.00 82	9.139 8	1.46 06	9.035 0	13.19 55	8.92 26	20.22 71	13.80 64	25. 012 2
क	0.66 33	7.98 60	3.33 70	3.77 14	7.983 3	10.00 82	9.139 8	1.46 06	9.035 0	13.19 55	8.92 26	20.22 71	13.80 64	25. 012 2
क	0.66 33	7.98 60	3.33 70	3.77 14	7.983 3	10.00 82	9.139 8	1.46 06	9.035 0	13.19 55	8.92 26	20.22 71	13.80 64	25. 012 2

VI. CLASSIFICATION

The classifiers are used to classify the features extracted. Each classifier will have its advantages and disadvantages depending on your particular application. This paper has chosen Probabilistic Neural Network (PNN) classifier to classify the features extracted and further to recognize the characters.

1. PNN CLASSIFIER

Probabilistic Neural Network (PNN) is often used in classification problems. It is a feed- forward neural network and is widely used in pattern recognition problems. Using PNN, the operations are arranged systematically into a multilayered feed-forward network with four layers [10]. In the presence of input, the 1st layer gives the distance from the input vector to the training input vectors. This result into a vector where its elements shows how close the input is to the training input. The 2nd layer gives the contribution for each class of inputs and result into the net output as a vector of probabilities. At last, using transfer function on the output of the 2nd layer, which is then, picks the maximum of these probabilities. The complete PNN is illustrated in figure 7. Thus, it produces a positive identification (1) and negative identification (0) for non targeted classes. The major advantage of using PNN over other classifiers like MLP (Multilayer Perceptron) is that PNN networks produces accurate predicted target probability scores [11].

If a Probabilistic Neural Network (PNN) is chosen, all the weights of the network can be calculated analytically. In this case, the number of cluster centers is by definition equal to the number of exemplars, and they all are set to the same variance (which may be optimized if a cross validation set is specified) [12].

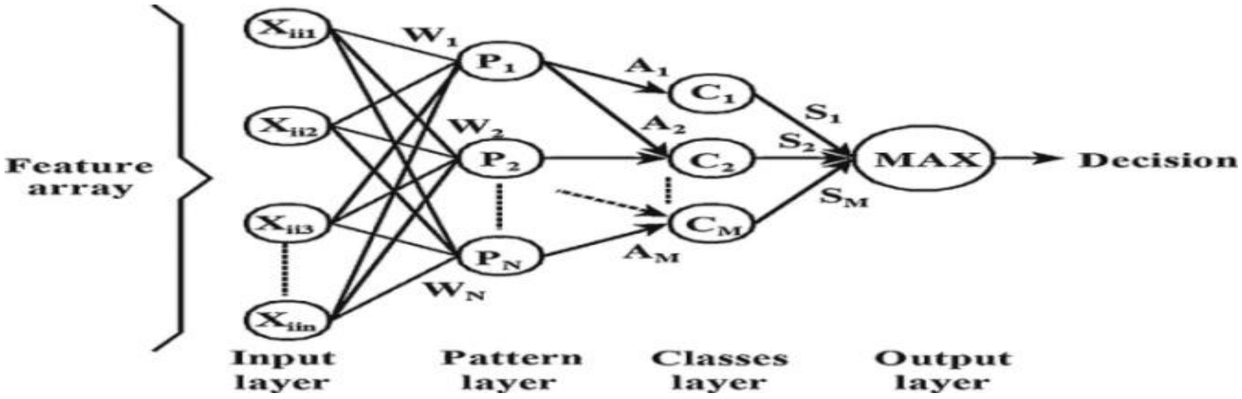


Fig.7: Probabilistic Neural Network (PNN).

VII. RESULT AND ANALYSIS

Character recognition is always a challenging and difficult task, because of variety in font size, font faces, distortions and noises applied to CAPTCHA code. So, the attempt was to achieve higher accuracy in character recognition. The result of classification using PNN is given in performance metrics 1 & 2:

Performance Metrics 1: Result of end-bar characters and digits using PNN classifier

Model Name: **Probabilistic Neural Network**

Training		
MSE	**r**	**Correct**
0.000944	0.986532	99.29%
Cross Validation		
MSE	**r**	**Correct**
0.003405	0.935513	96.86%
Testing		
MSE	**r**	**Correct**
0.001401	0.971967	96.28%
Training	Cross Val.	Testing

# of Rows	**5100**	**1275**	**2125**
MSE	**0.000944**	**0.003405**	**0.001401**
Correlation(r)	**0.986532**	**0.935513**	**0.971967**
# Correct	**5064**	**1235**	**2046**
# Incorrect	**36**	**40**	**79**
% Correct	**99.29%**	**96.86%**	**96.28%**

From the performance metrics 1, for the Devanagari characters with end-bar, the average accuracy of all, Training, Cross Validation and Testing is **97.48%.**

Performance Metrics2: Result of middle-bar, no-bar characters and digits using PNN classifier

Model Name: Probabilistic Neural Network	Training		
	MSE	**r**	**Correct**
	0.000144	0.998219	99.78%
	Cross Validation		
	MSE	**r**	**Correct**
	0.001069	0.990502	96.78%
	Testing		
	MSE	**r**	**Correct**
	0.001761	0.975215	96.07%
	Training	Cross Val.	Testing
# of Rows	**3600**	**900**	**1500**
MSE	**0.000144**	**0.001069**	**0.001761**
Correlation (r)	**0.998219**	**0.990502**	**0.975215**
# Correct	**3592**	**871**	**1441**
# Incorrect	**8**	**29**	**59**
% Correct	**99.78%**	**96.78%**	**96.07%**

From the performance metrics 2, for the Devanagari characters with middle-bar, no-bar and digits, the Average accuracy of all, Training, Cross Validation and Testing is **97.54%.**

Thus, using PNN classifier, for all 48 Devanagari characters the Average accuracy achieved is 97.51%. Ramteke and et al. [13] used ANN classifier and recorded an average recognition rate of 87.88% for Devanagari characters. Prof. Sheetal A. Nirve & Dr. G. S. Sable in their work [14], obtained recognition rate of all Devanagari characters is nearly up to 95% using ANFIS. It is observed from our experimental results that the proposed scheme, as compare to other techniques, has given the better average recognition rates for Devanagari characters.

VIII. FUTURE SCOPE

The proposed system used only limited Devanagari characters with middle-bar, no-bar and 10 numerals. This system can be extended to all Devanagari consonants and vowels along with modifiers. The proposed scheme hopefully can inspire a new thinking and innovative way to tackle the character recognition problem.

IX. REFERENCES

[1] . A.Abiya Jecinth kumar, A.R.Guru Gokul, “A Study On Captchas The Challenge Response Test”, International Journal of Latest Trends in Engineering and Technology Special Issue April-2018, pp. 005-011, 2018.

[2] . Dr. Rachoti Devaru, “Development of scripts in India-A Study “, Journal of Emerging Technologies and Innovative Research (JETIR), April 2020, Volume 7, Issue 4, 2020.

[3] . Prof. B. Mallikarjun, “Indian Model of Language Management “,Language in India, India’s Higher Education Authority UGC Approved List of Journals Serial Number 49042 , Vol. 18:11 November 2018.

[4] . Proposal for a Devanagari Script Root Zone Label Generation Rule-Set(JGR), LGR Version: 3.0 Date: 2019-03-06 Document version: 6.3 Authors: Neo-Brahmi Generation Panel [NBGP] , 2019.

[5] . Prafulla E. Ajmire, S.E. Warhede, “An Analytical Study of Devanagari Script Recognition”, In 61st IETE Annual Convention 2018 on “Smart Engneering for Sustainable Development”, Special Issue of IJECSCSE, ISSN: 2277-9477, 2018.

[6] . Anna Klappenbach, “The 12 most spoken languages in the world”, Article, January, 26, 2021, Source: https://blog.busuu.com/most-s, Access date: 07-10-2021.

[7] . Guidelines For Indian Government Websites (Revised edition, includes apps) An Integral Part of Central Secretariat Manual of Office Procedure, February 2018: Second Edition, Prepared By: National Informatics Centre (NIC), Ministry of Electronics & Information Technology (MeitY), Government of India, 2018

[8] . Anjali A. Chandavale, A. Sapkal, “Security analysis of CAPTCHA”, International Conference on Security in Computer Networks and Distributed Systems, Springer-Verlag Berlin Heidelberg, October 2012.

[9] . Mingqiang Yang, Kidiyo Kpalma, Joseph Ronsin, “A Survey of Shape Feature Extraction Techniques”, Peng-Yeng Yin. Pattern Recognition, IN-TECH, pp.43-90, 2008. Ffhal-00446037f, 2008.

[10] . Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara et. al, “Comprehensive Review of Artificial Neural Network Applications to Pattern Recognition”, IEEE, 2019.

[11] . P K Patra, M Nayak, S Kumar Nay and Nataraj Kumar Gobb, “Probabilistic Neural Network for Pattern Classification”, Proceedings of International Joint Conference on Neural Networks. IJCNN 2002.

[12] . Maciej Kusy, Piotr A. Kowalski “Modification of the Probabilistic Neural Network with the Use of Sensitivity Analysis Procedure”, Proceedings of the Federated Conference on Computer Science and Information Systems, pp. 97–103, Vol. 8., IEEE, 2016.

[13] . Surendra. P. Ramteke, Ramesh.D Shelke, Nilima P.Patil, “A Neural Network Approach to Printed Devanagari Character Recognition”, in International Journal of Computer Applications (0975 – 8887) Volume 61– No.22, January 2013.

[14] . Prof. Sheetal A. Nirve, Dr. G. S. Sable, “Optical character recognition for printed text in Devanagari using ANFIS”, International Journal of Scientific & Engineering Research, Volume 4, Issue 10, October-2013 236 ISSN 2229-5518.