ISSN: 2229-6956(ONLINE)
DOI: 10.21917/ijsc.2012.0041

COMPARISON OF SVM AND FUZZY CLASSIFIER FOR AN INDIAN SCRIPT

M. J. Baheti[1] and K. V. Kale[2]

[1]*Department of Computer Science and Engineering, Shri Neminath Jain Brahmacharyashram's Late Sau. Kantabai Bhavarlalji Jain College of Engineering, Maharashtra, India*
E-mail: mamtaji_61079@rediffmail.com

[2]*Department of Computer Science and Information Technology, Dr. Babasaheb Ambedkar Marathwada University, Maharashtra, India*
E-mail: kvkale91@gmail.com

Abstract

With the advent of technological era, conversion of scanned document (handwritten or printed) into machine editable format has attracted many researchers. This paper deals with the problem of recognition of Gujarati handwritten numerals. Gujarati numeral recognition requires performing some specific steps as a part of preprocessing. For preprocessing digitization, segmentation, normalization and thinning are done with considering that the image have almost no noise. Further affine invariant moments based model is used for feature extraction and finally Support Vector Machine (SVM) and Fuzzy classifiers are used for numeral classification. . The comparison of SVM and Fuzzy classifier is made and it can be seen that SVM procured better results as compared to Fuzzy Classifier.

Keywords:

Support Vector Machine, Fuzzy Classifier, Gujarati Handwritten Numerals

1. INTRODUCTION

Across the globe, almost more than 50 million people speak Gujarati, a language from Indo-Aryan family. In major, Gujarati is spoken and used as official language in Gujarat, a state in India. Irrespective of its wide popularity and use, Gujarati finds less documentation on recognition. It is derived from Devanagari and shares some appearances as that of Devanagari, Sanskrit, Marathi, etc. There is a wide variety in numerals in Indian languages. Fig.1 shows that the numerals belong to Gujarati language.

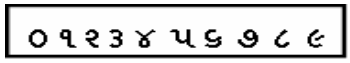

Fig.1. Gujarati Numerals 0 to 9

As mentioned earlier, Gujarati numerals show some same appearances like other numerals in Devanagari. Numerals like 0,2,3,4,7and 8 are same as that in Devanagari but numeral 1 has a bit tilt in the centre in Devanagari whereas it is straight line incase Gujarati 1. Numerals 0, 3, and 7 share confusion among Gujarati numerals while numerals 2 and 4 too are confusing. Confusion arises among 1 and 6. For numerals 1 and 5 confusion may arise due to closed loop for numeral 1 and open in case of numeral 5. Simultaneously numeral 8 and 9 share confusion in shapes.

This paper deals with the problem of recognition of Gujarati handwritten numerals. Gujarati numeral recognition requires performing some specific steps as a part of preprocessing. For preprocessing digitization, segmentation, normalization and thinning are done with considering that the image is having almost no noise. Further affine invariant moments based model is used for feature extraction and finally Support Vector Machine (SVM) and Fuzzy classifiers are used for numeral classification. This paper is organized in following sections; Section 2 describes brief literature survey done for Indian languages recognition. Section 3 details the steps taken for preprocessing. Section 4 describes algorithm which we have used to implement the paper. Section 5 elaborates the feature extraction done. Section 6 describes SVM and Fuzzy based numeral recognition. Section 7 details the conclusion of work done.

2. LITERATURE SURVEY

With the advent of technological era, conversion of scanned document (handwritten or printed) into machine editable format has attracted many researchers. Much work has been contributed with the continued effort for recognition of scripts in India. But less amount of work has been surveyed that addresses the recognition of Gujarati language. Although recognition of handwritten numerals is well researched topic but not much work has been reported on Gujarati handwritten numerals, in recent times.

The efforts for Gujarati character recognition started by the primitive effort of Antani and Agnihotri [1] in 1999 for printed characters. The authors used Euclidean and hamming distance classifiers for classification of various printed Gujarati characters. Dholakia [2] added his contribution in Gujarati character recognition by giving combined approach of wavelet feature extraction and neural net architecture to classify printed Gujarati characters. Desai [3] has reported recognition of Gujarati handwritten numerals employing skew correction, normalization and then direction profiles as feature vectors using neural net architecture to classify the numerals.

Devanagari got its primitive work in 1979 by Sinha and Mahabala [4]. They reported the structural characteristics of Devanagari script. Satish [5] studied Zernike moments and used it for Devanagari handwritten character recognition. Veena [6] described the method to describe the shapes of Devanagari characters and use them for recognition. Bhoumik et. al. [7] proposed an HMM based recognition scheme for handwritten Oriya numerals. Roy et al. [8] used chain code histogram contour points of the segmented numeral and applied neural network and quadratic classifier. Rao et al. [9] adopted feature based approach for isolated Telugu characters. Lakshmi et al. [10] addressed the recognition of printed basic symbols of Telugu language. They used seven moments for feature extraction and KNN as the classifier. Kurian et al. [11] has reported his effort for isolated Malayalam digit recognition using SVM. Jagadeesh Kannan [12] have fused HMM and SVM and used neural network to predict the correct character from Tamil script. Mahmud et al. [13] used free man chain code for scaled character and classified using feed forward neural network based recognition scheme.

3. PREPROCESSING

As there was no database available for Gujarati Script, we have created the database by taking the samples of handwritten imprints on that datasheets created to take the samples. Then these samples were scanned using HP 2400 Scanjet scanner at the resolution of 300dpi. The samples were withdrawn at random from various writers belonging to different profession, age, sex and education levels and were using different ink for preparing the samples on datasheets. Ten samples were taken from eight persons as shown in Fig.2. The database was created manually.

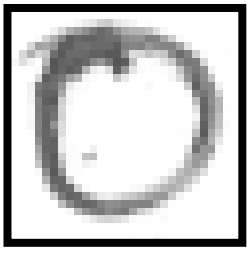

Fig.2. Original Handwritten Numeral Zero from Gujarati script stored in our database

Generally every implementation of script recognition requires a number of pre-processing steps followed by the actual recognition. The preprocessing steps vary in their deployment depending on age of the document, paper quality, resolution of the scanned image, the amount of skew in the image, the format and layout of the images and text, the kind of script used and also on the printed or handwritten characters. The actual recognition involves generally evaluation of statistical parameters finally employed to recognize the character. Typical preprocessing stages include noise removal, binarization, skeletonization, skew detection and correction, segmentation, etc.

3.1 IMAGE BINARIZATION

Binarization is a method which coverts rgb or gray scale images to binary images. The popular technique employed for binarization is threshold i.e. after selection of optimum threshold for the image convert all the intensity values above the threshold intensity to one intensity value representing either “black” or “white” value. We have used Otsu’s threshold algorithm to get binary image as shown in Fig.3.

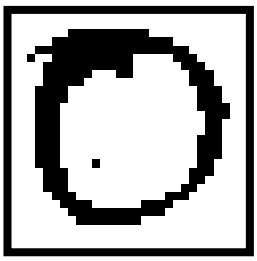

Fig.3. Binarized image of Numeral zero from Gujarati script

3.2 NOISE REMOVAL

Generally noise comes in the documents due to dust or spread of ink on printer, scanner, age of the document, etc. So, it is required to remove this noise before it is subjected to segmentation or any other process. The most popular technique is to use low-pass filter on the binary image. We have not used any noise removal technique for our algorithm as the binary images that were considered were abstracted from good quality scanned document and use it for later processing.

3.3 SEGMENTATION

The next stage is segmenting the document into its sub components. Segmentation is an operation that seeks to decompose an image of sequence of characters into sub images of individual symbols. Character Segmentation strategies are considered on two fronts.

- Segmentation by connected component analysis
- Recognition based segmentation

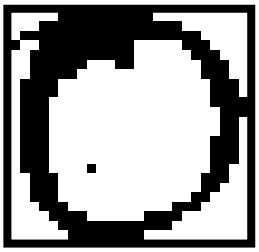

Fig.4. Segmentation by connected component analysis

3.4 NORMALIZATION

Due to variation in writing styles one has to employ the algorithm so that there is uniformity in the input numerals. For this reason segmented numerals are scaled up or scaled down to standard size window of 40 X 40 as shown in Fig.5. Every measure has to be taken to preserve the exact aspect ratio of the input numeral.

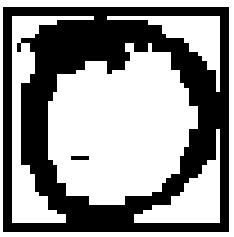

Fig.5. Size Normalization of Gujarati Numeral zero of size 40x40

3.5 SKELETONIZATION

Due to normalization, the image is either scaled up or scaled down, resulting in addition of pixels in the image. To manage the numeral irrespective of the shape, the skeleton of the image is found. Then this skeletonized image as shown in Fig.6 is put forward for feature extraction.

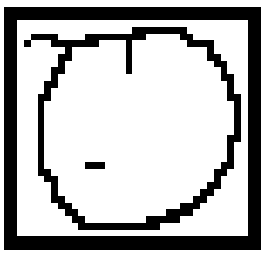

Fig.6. Single pixel skeletonized image of Gujarati numeral zero

4. ALGORITHM

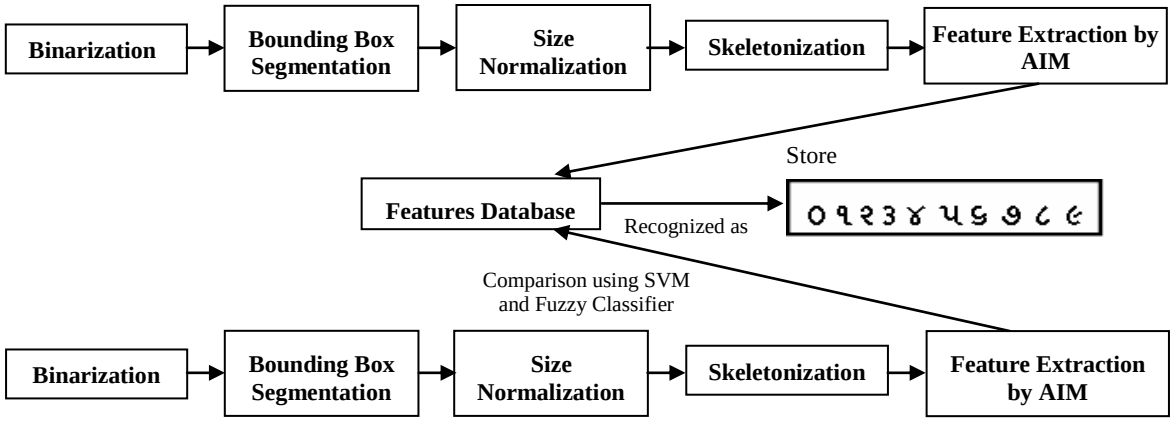

Fig.7. Schematic block diagram of Gujarati Numerals

The algorithm (as shown in Fig.7) identifies the handwritten Gujarati numerals based on iterative approach. It identifies more than one numeral. The steps are elaborated in two sets viz., training and testing.

Training:

- Binarization of the scanned document of Gujarati numerals.
- Labeling of existing connected components.
- Perform segmentation for individual labeled component with bounding box.
- Normalize each segmented component to a standard size of 40x40 pixels.
- Skeletonized the image of the numeral so that it is one pixel wide.
- Extract feature vector for each numeral given as input and store it in the train database.
- Find the mean and standard deviation of the features set.

Testing:

- Binarization of the scanned document of Gujarati numerals.
- Labeling of existing connected components.
- Perform segmentation for individual labeled component with bounding box.
- Normalize each segmented component to a standard size of 40x40 pixels.
- Skeletonized the image of the numeral so that it is one pixel wide.
- Extract feature vector for each numeral given as input and store it in the test database.
- Train the SVM using train database.
- Test the SVM as well as fuzzy classifier separately using test database.
- For each numeral compute the matching probabilities. The most likely probability is found and maximum probabilities are outputted. The numeral which has the highest probability matched is being displayed as recognized numeral.

The overall recognition rate if finally achieved by ratio of sum of the correctly recognized numerals by total numerals for SVM as well as fuzzy classifier.

5. FEATURE EXTRACTION

The image of all the segmented characters is normalized to a common height and width producing a grid of 40 X 40 pixel size. This normalized image is then thinned out so that it is one pixel wide. Now the affine invariant moments are derived for each of the numeral image as follows.

Flusser and Suk [20] have derived a set of moment invariants, which are invariant under affine transformation. These affine moment invariants [AMIs] represent a significant contribution to the progress in the field of invariant pattern recognition. The AMIs is invariant under general affine transformation,

$$\begin{aligned} u &= a_0 + a_1 x + a_2 y \\ v &= b_0 + b_1 x + b_2 y \end{aligned} \tag{1}$$

where, (x, y) and (u, v) are coordinates in the image plan before and after the transformation respectively. The basic affine invariant moments are given below:

$$\begin{aligned} I_1 &= (\mu_{20}\mu_{02} - \mu_{11}^2)/\mu_{00}^4 \\ I_2 &= (\mu_{30}^2\mu_{03}^2 - 6\mu_{30}\mu_{21}\mu_{12}\mu_{03} + 4\mu_{30}\mu_{12}^3 + 4\mu_{03}\mu_{21}^3 \\ &\quad - 3\mu_{21}^2\mu_{12}^2)/\mu_{00}^{10} \\ I_3 &= (\mu_{20}(\mu_{21}\mu_{03} - \mu_{12}^2) - \mu_{11}(\mu_{30}\mu_{03} - \mu_{21}\mu_{12}) \\ &\quad + \mu_{02}(\mu_{30}\mu_{12} - \mu_{21}^2))/\mu_{00}^7 \\ I_4 &= (\mu_{20}^3\mu_{03}^2 - 6\mu_{20}^2\mu_{11}\mu_{12}\mu_{03} - 6\mu_{20}^2\mu_{02}\mu_{21}\mu_{03} + \\ &\quad 9\mu_{20}^2\mu_{02}\mu_{12} + 12\mu_{20}\mu_{11}^2\mu_{21}\mu_{03} + 6\mu_{20}\mu_{11}\mu_{02}\mu_{30}\mu_{03} - \\ &\quad 18\mu_{20}\mu_{11}\mu_{02}\mu_{21}\mu_{12} - 8\mu_{11}^3\mu_{30}\mu_{03} - 6\mu_{20}\mu_{02}^2\mu_{30}\mu_{12} + \\ &\quad 9\mu_{20}\mu_{02}^2\mu_{21} + 12\mu_{11}^2\mu_{02}\mu_{30}\mu_{12} - 6\mu_{11}\mu_{02}^2\mu_{30}\mu_{21} \\ &\quad + \mu_{02}^3\mu_{30}^2)/\mu_{00}^{11} \end{aligned} \tag{2}$$

6. SVM AND FUZZY BASED RECOGNITION

6.1 SVM CLASSIFIER

Support vector machine [14] [15] [16] is new classifier that is extensively used in many pattern recognition applications. On pattern classification problem, SVM demonstrate very good generalization performance in practical applications. SVM are binary classifiers that separate linearly any two classes by finding a hyper plane of maximum margin between the two classes. The margin means the minimal distance from the separating hyper plane to the closest data points. SVM learning machines searches for an optimal separating hyper plane, where the margin is maximal. The outcome of the SVM is based only on the data points that are at the margin and called support vectors.

There are two approaches to extend SVMs for multi-class classification. First one is one against one (ONO) and other is one against all (ONA). We have used ONA approach where N classifiers are performed to separate one of N mutually exclusive classes from all other classes. An SVM assumes that all samples in the training set are identically distributed and independent. A kernel is utilized to map the input data to a higher dimensional feature space so that the problem becomes linearly separable. The kernel plays a very important role. Gaussian kernel performs superior compare to linear kernel, polynomial kernel etc. We have used Gaussian kernel.

We have collected hand written Gujarati numerals of a number of people. Following the same preprocessing procedure of an input image, we have collected the 10 Gujarati numerals of different handwriting. After getting input from affine invariant moments for each numeral image, the SVM tries to find the class where the input numeral image should belong.

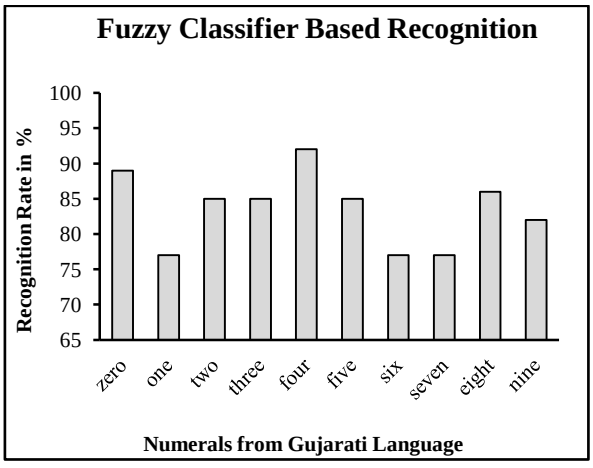

Fig.8. Recognition Rate using SVM classifier

6.2 FUZZY CLASSIFIER

Before training the features set for fuzzy classifier we need to find the mean and standard deviation of the features set.

$$\text{Mean } M_i = \frac{1}{N_i}\sum I_{i(k)} \tag{3}$$

$$\text{Std-Dev } \sigma_i = \sqrt{\sum (I_{i(k)} - M_i)^2} \tag{4}$$

where, N_i is the number of samples in i^{th} class and $i_{(k)}$ Stands for the k^{th} feature value of reference numeral in the i^{th} class.

For an unknown input numeral *x*, the features are extracted using the affine invariant moments model. The membership function is chosen as,

$$\mu_i = \exp(x_i - M_i)^2 / 2\sigma_i^2 \tag{5}$$

where, x_i is the i^{th} feature of the unknown numeral.

If all *xi*'s are close to μ_i which represent the known statistics of a reference character, then the unknown numeral is identified with this known numeral because all membership function values are close to 1 and hence the average membership function is almost 1 [19].

Let, $M_i(r)$ and $\sigma_i^2(r)$ belong to the r^{th} reference numeral with r = 0,1…9, we then calculate the average membership as,

$$\mu_{av}(r) = 1/c\sum_{i=1}^{c} \exp(x_i - M_i)^2 / 2\sigma_i^2 \tag{6}$$

where, *c* denotes for the number of fuzzy sets. Then $x \epsilon r$ if $\mu_{av}(r)$ is the maximum for r = 0,1…9.

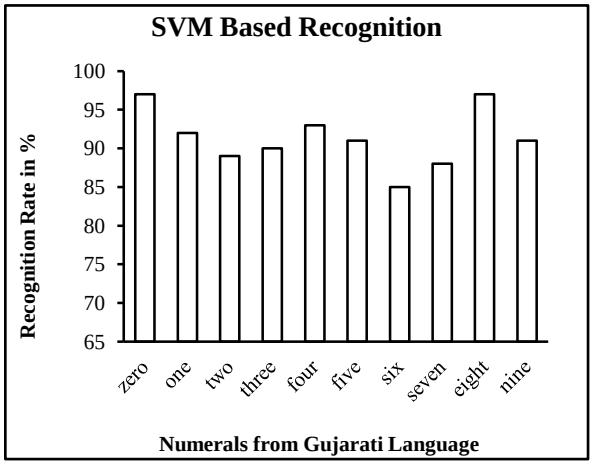

Fig.9. Recognition rate using Fuzzy Classifier

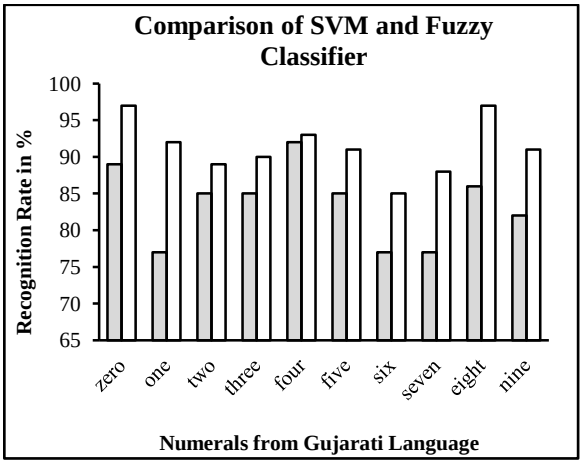

Fig.10. Comparison of SVM and Fuzzy classifier

Numerals 1, 6 and 7 have shown less recognition rate in case of fuzzy classifier as compared to others. Numerals 8, 4 and 0 show optimum results for the fuzzy classifier (as shown in Fig. 9). When we see the recognition graph of SVM (from Fig. 8), we find that SVM has shown good results for numerals 0,1,4,8, and 9 while other numerals show less rate of recognition but more better than that shown by fuzzy classifier. Also as compared to overall recognition rate, SVM has shown recognition rate of 91.25% where as fuzzy classifier shows 83.45%. One can observe from Fig.10, in case of SVM as well as Fuzzy Classifier numerals 6 and 7 have less recognition rate as compared to other numerals.

But among SVM [23] and Fuzzy Classifier, SVM shows good results as compared to Fuzzy Classifier. Best results are seen for numerals 0, 8, 4, and 1. Moreover these results were compared with the recognition rates obtained by Desai [3] and Prasad [21-22] and were found to be better.

7. CONCLUSION

Recognition rate is highly affected by similarity of various numerals. The numerals used in this experiment are enclosed in a bounding region of a fixed size. These bounded numerals are then skeletonized. We have derived affine invariant moments as features set and compared these features using SVM and Fuzzy Classifier. By doing comparison of SVM and Fuzzy classifier it has been seen that SVM procured better results as compared to Fuzzy Classifier. For affine invariant moments SVM has produced 91.25% of recognition rate whereas fuzzy classifier has produced as compared to SVM a meager amount of recognition accuracy of 83.45%.

REFERENCES

[1] Antani S and Agnihotri L, "Gujarati Character Recognition" *Proceedings of the 5th International Conference on Document Analysis and Recognition (ICDAR'99)*, pp. 418-422, 1999.

[2] Dholkia J, Yajnik A and Negi A, "Wavelet Feature Based Confusion character sets for Gujarati script", *Proceedings of the International Conference of Computational Intelligence and Multimedia Application (ICCIMA'07)*, Vol. 02, pp. 366-371, 2007.

[3] Desai A A, "Gujarati handwritten numeral optical character reorganization through neural network", *Pattern Recognition*, Vol. 43, pp. 2582-2589, 2010.

[4] Sinha R.K and Mahabala, "Machine recognition of Devnagari script", *IEEE Transactions on Systems, Man, and Cyber*netics, Vol. 9, No. 8, pp. 435–441, 1979.

[5] Kumar S and Singh C, "A Study of Zernike Moments and its use in Devanagari Handwritten Character Recognition", *Proceedings of the International Conference on Cognition and Recognition(ICCR'05)*, pp. 514-520, 2005.

[6] Bansal V, "Integrating knowledge sources in Devnagari text recognition", *Ph.D. Thesis, IIT Kanpur*, 1999.

[7] Bhowmik T. K, Parui S. K, Bhattacharya U and Shaw B, "An HMM Based Recognition Scheme for Handwritten Oriya Numerals" *Proceedings of the 9th International Conference on Information Technology (ICIT'06)*, pp. 105-110, 2006.

[8] Rao P and Ajitha T, "Telugu script recognition", *Proceedings of the Third International Conference on Document Analysis and Recognition (ICDAR'95)*, Vo1. 1, pp. 323–326, 1995.

[9] Roy K, Pal T, Pal U and Kimura F, "Oriya Handwritten Numeral Recognition System" *Proceedings of the Eighth International Conference on Document Analysis and Recognition (ICDAR'05)* pp. 770-774, 2005.

[10] Lakshmi C. V and Patvardhan C, "Optical Character Recognition of Basic Symbols in Printed Telugu Text" *IE(I)Journal-CP*, Vol. 84, pp. 66-71, 2003.

[11] Cini Kurian, Firoz Shah. A and Kannan Balakrishnan, "Isolated Malayalam Digit Recogntion Using Support Vector Machines", *IEEE International Conference on Communication Control and Computing Technologies*, pp. 692-695, 2010.

[12] Kannan R. J and Prabhakar R, "Accuracy Augmentation of Tamil OCR using Algorithm Fusion", *International Journal of Computer Science and Network Security*, Vol. 8, No. 5, pp. 51-56, 2008.

[13] Mahmud J. U, Mohammed F. R and Chowdhury M. R, "A Complete OCR System for Continuous Bengali Characters", *Conference on Convergent Technologies for Asia-Pacific Region Tencon 2003*, Vol. 4, pp. 1372 – 1376, 2003.

[14] Vapnik V, "*The Nature of Statistical Learning Theory*", Springer Verlag. 1995.

[15] Burges C, "A Tutorial on support Vector machines for pattern recognition", *Data mining and knowledge discovery*, Vol. 2, pp. 1-43, 1998.

[16] Vapnik V. N, "*Statistical Learning Theory*", John Wiley and sons. 1998.

[17] Hall P, Park B.U and Samworth R.J, "Choice of neighbor order in nearest neighbor classification", *Annals of Statistics*, Vol. 36, No. 5, pp. 2135-2152, 2008.

[18] Cover T.M and Hart P. E, "Nearest neighbor pattern classification", *IEEE Transactions of Information Theory*, Vol. 13, No. 1, pp. 21-27, 1967.

[19] Hanmandlu M and Murthy O.V.R, "Fuzzy Model Based Recognition of Handwritten Numerals", *Journal on Pattern Recognition*, Vol. 40, No. 6, pp. 1840-1854, 2007.

[20] Flusser J and Suk T, "Affine Moment Invariants: A new tool for Character Recognition", *Pattern Recognition Letters*, Vol. 15, pp. 433-436, 1994.

[21] Prasad J.R, Kulkarni U.V and Prasad R.S, "Template Matching Algorithm for Gujarati Character Recognition", *IEEE International Conference on Emerging Trends in Engineering and Technology*, pp. 263 – 268, 2009.

[22] Prasad J.R, Kulkarni U.V and Prasad R.S, "Offline Handwritten Character Recognition of Gujarati script using Pattern Matching", *3rd International Conference on Anti-counterfeiting, Security, and Identification in Communication*, pp. 611 – 615, 2009.

[23] Baheti M.J, Mane A.V, Hannan M.S and Kale K.V, "Comparison of PCA and SVM for a West Indian Script-Gujarati," *CIIT International Journal of Digital Image Processing*, Vol. 3, No. 11, pp. 709-715, 2011.