k-NN based On-Line Handwritten Character recognition system

Sreeraj.M
Department of Computer Science
Cochin University of Science and Technology
Cochin, India
msreeraj@cusat.ac.in

Sumam Mary Idicula
Department of Computer Science
Cochin University of Science and Technology
Cochin, India
sumam@cusat.ac.in

Abstract—**On-line handwriting recognition has been a frontier area of research for the last few decades under the purview of pattern recognition. Word processing turns to be a vexing experience even if it is with the assistance of an alphanumeric keyboard in Indian languages. A natural solution for this problem is offered through online character recognition. There is abundant literature on the handwriting recognition of western, Chinese and Japanese scripts, but there are very few related to the recognition of Indic script such as Malayalam. This paper presents an efficient Online Handwritten character Recognition System for Malayalam Characters (OHR-M) using K-NN algorithm. It would help in recognizing Malayalam text entered using pen-like devices. A novel feature extraction method, a combination of time domain features and dynamic representation of writing direction along with its curvature is used for recognizing Malayalam characters. This writer independent system gives an excellent accuracy of 98.125% with recognition time of 15-30 milliseconds.**

Keywords-On-Line character recognition, Malayalam, k-NN, Feature extraction

I. INTRODUCTION

The fast evolving electronic gadgets of today vie with one another in providing faster, better and friendlier user interactions. Intuitive and simple user interfaces encourage people of diverse backgrounds and tastes to interact with any kind of system without any hesitation. Evolution of computers during the last few decades has been phenomenal both in hardware and software components. The major thrust in the development has been the enhancement of computer interfaces by way of both hardware and software innovations. Various new and innovative digitizing and motion capturing devices are being added to the computer interface. The development of pen interfaces is such a key element in providing an efficient and natural way of input to the computer. For example PDAs usually have a graphical user interface in which a pen can be used for pointing and selecting functions, drawing, and text entry. Human beings use natural language as the primary mode of communication. The users of computer demand the need for written as well as spoken natural language. This is a requirement in India as it is a multi-lingual and multi-script country comprising of many official languages namely Assamese, Bengali, Bodo, Dogri, Gujarati, Hindi, Kannada, Kashmiri, Konkani, Maithili, Malayalam, Manipuri, Marathi, Nepali, Oriya, Punjabi, Sanskrit, Santhali, Sindhi, Tamil, Telugu and Urdu. Recognition of handwritten Indian scripts is difficult because of the presence of numerals, vowels, consonants, vowel modifiers and compound characters. The structure of the scripts and the variety of shapes and writing styles pose challenges that are different from other scripts and hence require customized techniques for feature representation and recognition.

There are two modes of character recognition: online and offline. In online mode, a user writes directly on an electronic surface with a stylus or pen. The trajectory of the pen is electronically recorded in the form of a sequence of (x,y) coordinate pairs as a function of time. The additional temporal information available in online mode leads to better recognition performance. Off-line handwriting recognition is the process of recognizing characters and words present in the digital images of the handwritten text. Considerably less work has been done towards handwritten character recognition of Indian languages than for other languages. Some effort for the offline recognition of Malayalam characters are reported [1] [2], but very little effort has been reported for the on-line recognition. In this paper we describe a k-NN based scheme for the recognition of online handwritten characters of popular south Indian script, Malayalam.

The System described in this paper is developed using Java. The system has five major modules namely Data acquisition and Preprocessing, Feature extraction, Training, Recognition and intellisence and manipulation of Conjunct Characters. The system is first trained with the character set before using as recognizer.

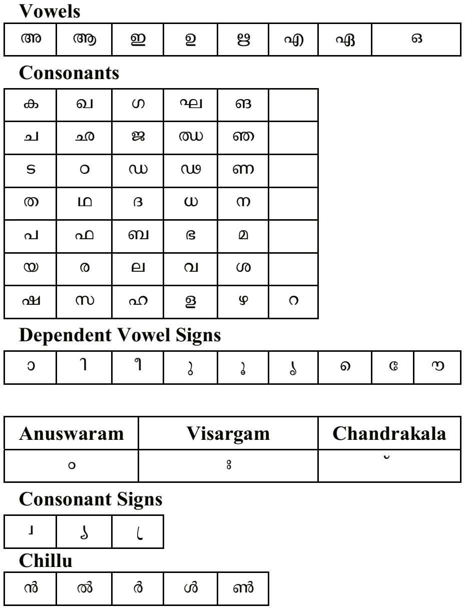
The organization of the paper is as follows. Section 2 describes Malayalam script characteristics. Section 3 gives the overview of the system architecture, feature extraction and learning algorithm. Section 4 gives the implementation and performance analysis and section 5 represents the concluding remarks.

978-0-7695-4152-5/10 $26.00 © 2010 IEEE
DOI 10.1109/ICIIC.2010.58

II. MALAYALAM SCRIPT CHARACTERISTICS

Malayalam is a Dravidian language spoken by about 35 million people. It is spoken mainly in the state of Kerala and in the Lakshadweep Islands.

Malayalam scripts have the following features. It has syllabic alphabet in which all consonants have an inherent vowel. Diacritics can appear above, below, before or after the consonant they belong to, are used to change the inherent vowel. When they appear at the beginning of a syllable, vowels are written as independent letters. When certain consonants occur together, special conjunct symbols are used which combine the essential parts of each letter. There are about 128 characters in the Malayalam alphabet which includes vowels (15), consonants (36), chillu (5), anuswaram, visargam, chandrakkala-(total-3), consonant signs (3), left vowel signs (2), right vowel signs (7), conjunct consonants (57). Out of all these characters mentioned, only 64 of them are considered to be the basic ones as shown in Figure 1.



Vowels

അ	ആ	ഇ	ഉ	ഋ	എ	ഏ	ഒ

Consonants

ക	ഖ	ഗ	ഘ	ങ	
ച	ഛ	ജ	ഝ	ഞ	
ട	ഠ	ഡ	ഢ	ണ	
ത	ഥ	ദ	ധ	ന	
പ	ഫ	ബ	ഭ	മ	
യ	ര	ല	വ	ശ	
ഷ	സ	ഹ	ള	ഴ	റ

Dependent Vowel Signs

ാ	ി	ീ	ു	ൂ	ൃ	െ	േ	ൗ

Anuswaram	Visargam	Chandrakala
ം	ഃ	്

Consonant Signs

്യ	്വ	്ര

Chillu

ൻ	ൽ	ർ	ൾ	ൺ

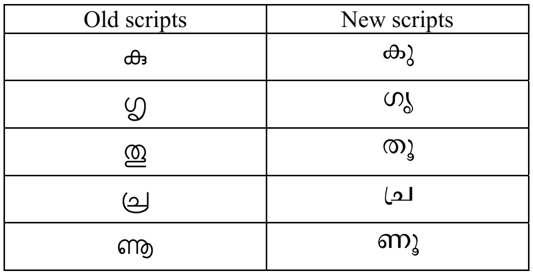
Figure 1. 64 basic characters of Malayalam

The properties of Malayalam characters are the following

- Since Malayalam script is an alphasyllabary of the Brahmic family they are written from left to right.
- Almost all the characters are circular by themselves. They consist of loops and curves. The loops are written frequently in the clockwise order.
- Several characters are different only by the presence of curves and loops.
- Unlike English, Malayalam scripts are not case sensitive and there is no cursive form of writing.
- Malayalam is a language which is enriched with vowels, consonants and has the maximum number of sounds that are not available in many other languages as shown in Figure 2.
- Two prominent ways of writing Malayalam scripts exists today. One followed by older generation and the other followed by younger generation. But the latter has become standard form even though usage of the former is still common. Some samples are given in Figure 3[3].

ണ, ള, ഴ, റ, ഓ, ഏ, ൾ, ൺ, ർ

Figure 2. Rare Sounds of scripts available only in Malayalam language.



Old scripts	New scripts
കു	കു
ഗൂ	ഗൂ
തൂ	തൂ
പ്ര	പ്ര
ഞൂ	ഞൂ

Figure 3. Old and new scripts of Malayalam.

III. SYSTEM OVERVIEW

The system for on-line recognition of Malayalam characters is developed in JAVA using K-NN classifier algorithm. Figure 4 gives the schematic representation of the system.

There are 57 numbers of conjunct letters in Malayalam which are derived from the basic 64 characters (Figure 1). When all the characters including the conjunct letters are included for the training sample it increases the complexity of the entire system. Efforts are put forward for the manipulation of conjunct characters using a rule based approach. A separate module named conjugator is incorporated with the recognizer for the same. Apart from this with extra assistance of intellisence for the benefit of the user is also included.

A. Pen device & Data sets

The raw data is collected using the device Wacom Graphire 4 CTE-640. Standards are important for the creation of handwriting datasets to ensure that resources created can be used by others. UNIPEN is still the de-facto standard for encoding of handwriting data because of its simplicity and widespread usage [4], [5]. For the attainment of maximum accuracy, a database of 40 writers consisting of 64 basic characters have been collected and trained. Then the test has been done by different schemes based on writer dependent and writer independent strokes.

The system consists of preprocessing feature extraction, training and recognition modules.

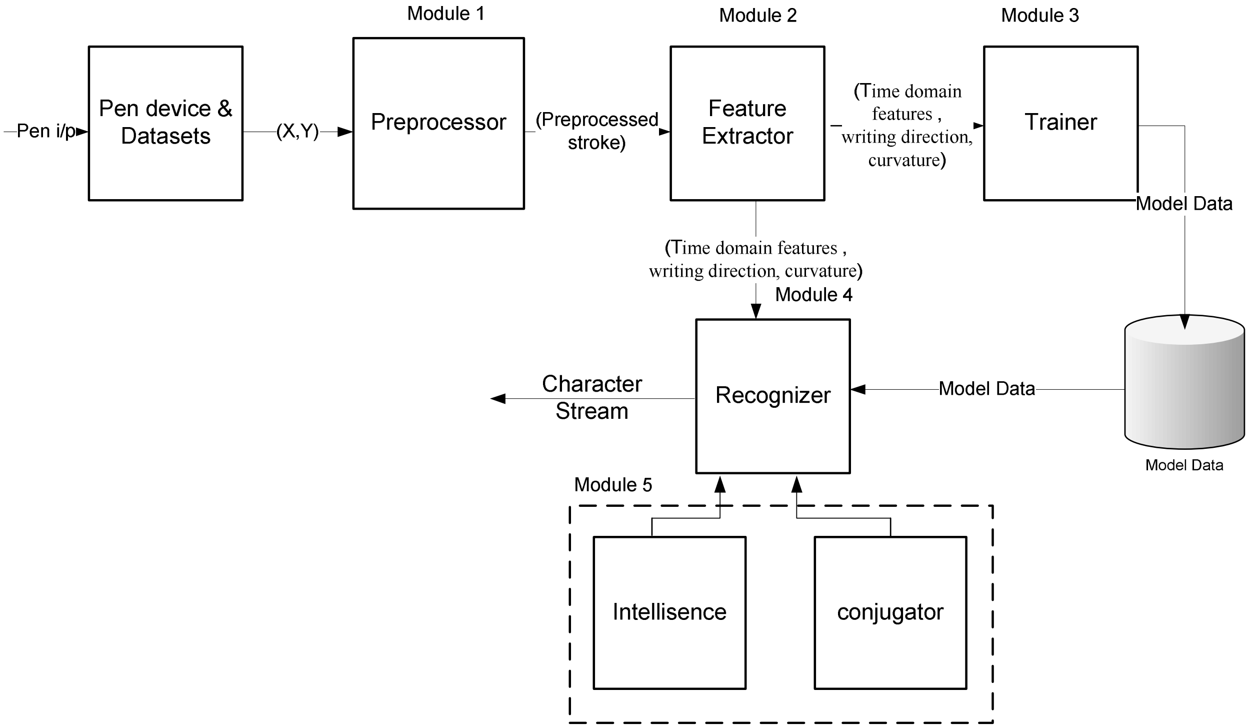


Figure 4. System architecture.

B. Preprocessing

Prior to any recognition, the captured data is generally preprocessed. It is for reducing spurious noise, normalizing the various aspects of the trace and segmenting the signal into meaningful units [6]. Preprocessing consists of the following steps.

1) Dot detection. Any stroke, if it has to be considered as a dot should be below the normalized dot size threshold. The threshold value is 0.01 and it is expressed in real length terms (inches) and converted internally to points using the knowledge of the device's spatial resolution. If the width and height is both less than this threshold then it treated as a dot.

2) Dehooking. Dehooking is done to eliminate stray strokes that appear due to inaccuracies in pen down position or rapid erratic motion in placing the stylus on the tablet. Strokes are detected by comparing the number of points with a threshold value. The mark is retained if the value is greater than the threshold value or removes it otherwise. A threshold value of length of 0.13 was chosen for dehooking process.

3) Smoothing. The strokes are smoothened using a larger filter so that unwanted cusps and intersections are removed and we obtain a smooth curve with lesser number of points. Since Malayalam characters are curvaceous in nature, smoothening of curves is essential for recognition.

To remove jitter from the handwritten text, we replace every point ($x(t), y(t)$) in the trajectory by the mean value of its neighbors:

$$x'(t) = \frac{x(t-N) + \ldots + x(t-1) + \alpha x(t) + x(t+1) + \ldots + x(t+N)}{2N + \alpha}$$

And

$$y'(t) = \frac{y(t-N) + \ldots + y(t-1) + \alpha y(t) + y(t+1) + \ldots + y(t+N)}{2N + \alpha}$$

The parameter α is based on the angle subtended by the preceding and succeeding curve segment of ($x(t)$, $y(t)$) and is empirically optimized. This helps to avoid the smoothing of sharp edges, which provide important information when there is a sudden change in direction [7].

4) Thinning. The user may pause in the midst of writing. Thinning is the removal of duplicated points during the process of writing.

5) Loop detection is a parameter to make out whether a stroke is a loop or not considering the value of a loop threshold.T

6) Normalization. Scaling is the next stage in preprocessing. Handwritten characters have different sizes necessitating normalization. Scaling involves the process of converting all characters to the predefined constant width and height while preserving the aspect ratio. This ensures that each handwritten characters have a canonical representation, so that the size makes no difference in recognition.

7) Orientation normalization is the process by which all strokes, regardless of the direction of writing, are normalized to go in a certain standard direction.

8) Equidistant Resampling. This resamples each stroke at equal intervals in space along its trajectory, and removes speed variations while writing across the writers. The resampling is performed such that a constant number of points are obtained from any trace [8].

Finally, the preprocessed stroke is further used in feature extraction. The resulting character is sharp and of standard size. This makes training and recognition phase more efficient and accurate.

C. Feature Extraction

This is the module where the features of handwritten characters are analyzed for training and recognition which are explained below. In proposed approach, each point on the strokes with values of selected features (time-domain features [9], [10] writing direction and curvature) is described in the consecutive sub sections.

1) Normalized x-y coordinates: The x and y coordinates from the normalized sample constitute the first 2 features.

2) Pen-up/pen-down: In this system the entire data is stored in UNIPEN format where the information of the stroke segments are exploited using the penup/pen-down feature. Penup/pen-down feature is dependent on the position sensing device. The pen-down gives the information about the sequence of coordinates when the pen touches the pad surface. The pen-up gives the information about the sequence of coordinates when the pen not touching the pad surface.

3) ***Aspect:*** Aspect at a point characterizes the ratio of the height to the width of the bounding box containing points in the neighborhood.

It is given by

$$A(t) = \frac{2 \times \Delta y(t)}{\Delta x(t) + \Delta y(t)} - 1$$

Where $\Delta x(t)$ and $\Delta y(t)$ are the width and the height of the bounding box containing the points in the neighborhood of the point under consideration. In this system, we have used neighborhood of length 2 i.e. two points to the left and two points to the right of the point along with the point itself.

4) Curvature: The curvature at a Point (x(n),y(n)) is represented by cosφ(n)a and sinφ(n). It can be computed using the following formulae [9].

$$Sin\ \phi(n) = Cos\ \theta(n-1) \times Sin\ \theta(n+1) - Sin\ \theta(n-1) \times Cos\ \theta(n+1)$$
$$Cos\ \phi(n) = Cos\ \theta(n-1) \times Cos\ \theta(n+1) + Sin\ \theta(n-1) \times Sin\ \theta(n+1)$$

5) Writing direction: The local writing direction at a point (x(n),y(n)) is described using the cosine and sine [10]:

$$\sin\theta\ (n) = \frac{Y_n - Y_{n-1}}{\sqrt{(X_n - X_{n-1})^2 + (Y_n - Y_{n-1})^2}}$$

$$\cos\theta(n) = \frac{X_n - X_{n-1}}{\sqrt{(X_n - X_{n-1})^2 + (Y_n - Y_{n-1})^2}}$$

The above elements will be the feature vector for training and recognition modules.

D. The Nearest Neighbor Classifier (K-NN)

K-NN is a type of instance-based learning where the function is only approximated locally and all computation is deferred until classification. In pattern recognition, the k-nearest neighbors algorithm (k-NN) is a method for classifying objects based on closest training examples in the space. The k-nearest neighbor algorithm is amongst the simplest of all machine learning algorithms. The algorithm computes the distance (or similarity) between each test sample and all the training samples to determine its nearest-neighbor list. Given a set of prototype vectors, $T_{XY} = \{(x_1, y_1), (x_2, y_2), \ldots\ldots\ldots\ldots\ldots, (x_i, y_i)\}$, the input vectors being $x_i \in X \subseteq R^n$ and corresponding targets being

$y_i \in Y = \{1, 2, \ldots\ldots\ldots\ldots, c\}$, let $R^n(x) = \{x' : \| x - x' \| \leq r^2\}$ be a ball centered in the vector x in which K prototype vectors

$x_i, i \in \{1, 2, \ldots\ldots\ldots\ldots, l\}$ lie,

i.e. $| x_i : x_i \in R^n(x) | = K$ .The k-nearest neighbor classification rule $q = X \rightarrow Y$ is defined as

$q(x) = \arg\max v(x, y)$, where $v(x, y)$ is the number of prototype vectors x_i with targets $y_i = Y$, which lie in the ball $x_i \in R^n(x)$.

E. Recognizer

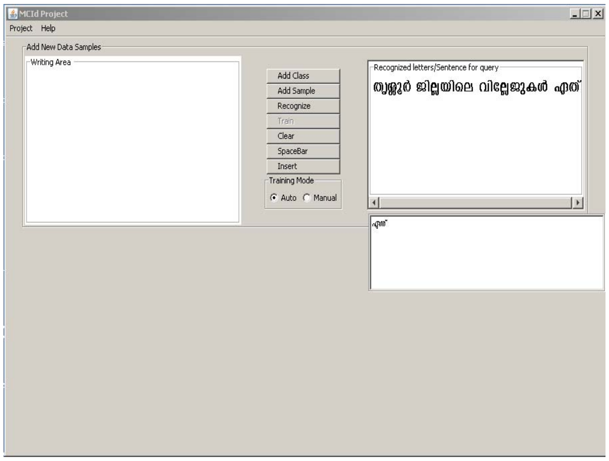
The feature extraction module resulted in a combination of time domain features and dynamic representation of writing direction along with its curvature. In the training module feature vectors and class labels of the training samples are stored as a model file. In the recognition module, the same features as before are computed for the test samples. Recognition function predicts the class label of the test sample by finding the Euclidean distance of the test sample to the classes according to the k-Nearest-Neighbor rule. Then the classification and recognition is achieved on the basis of similarity measurement.

IV. Implementation

The system was implemented in JAVA, on a 32-bit AMD Athlon 2.0 with 512MB RAM. A total of 2560 samples collected from 40 people for the system. The implementation includes coding of the various modules namely Data acquisition and pre-processing, Feature extraction, Training, Recognition with intellisence and manipulation of Conjunct Characters as described in the previous sections. The user data is collected using the device Wacom Graphire 4 CTE-640 and stored in UNIPEN format in a text file. For implementing the pre-processing module a property file containing the parameters necessary for pre-processing stages was created. This property file contains parameters like DotThreshold, DehookThreshold, LoopThreshold, PreserveAspectRatio, ResampDiamension, PrototypePerClass and ProtypeDistance for Dotdetection, Dehooking, Loopdetection, Normalization, Resampling, Prototype selection and distance measured for clustering respectively. Next feature vectors for every re-sampled point of each character is computed and passed as inputs for training and recognition modules.

For training, a list file is prepared by mapping all sample strokes with its class labels. This list file consists of all sample strokes of all characters. Prototype selection based on Hierarchical Clustering is performed on samples of each class and stored as the model file that will be used for recognition.

In the recognizer, list file prepared while training, property file and model file are loaded. The feature vector of the test sample is calculated and the nearest class label is recognized. The recognized class label is mapped to the corresponding Malayalam Unicode. This is displayed on the output screen. The output is analyzed for conjunct letters and replaced by a Rule based approach. To incorporate intellisence feature a dictionary with 2000 Malayalam words was created which will be loaded while recognition. A pop-up window displaying Malayalam words from the dictionary using a fast searching algorithm was implemented. Figure 5 shows the screen shot of recognized character stream.



Recognized letters/Sentence for query

തൃശ്ശൂർ ജില്ലയിലെ വില്ലേജുകൾ എത്

എത്

Figure 5. Screen shot of recognized Character stream

A. Performance Analysis

The success of handwritten recognition system is vitally dependent on its acceptance by potential users. The system was tested according to two schemes such as writer dependent and writer independent. Writer dependent testing was named Scheme 1 while writer independent testing was named as Scheme 2. Writer-dependent system provides a higher level of recognition accuracy than writer independent system. The amount of training data that must be supplied by the user before the system can be used may impede its acceptance in writer dependent system. On the other hand, a writer-independent system must be able to recognize a wide variety of writing styles in order to satisfy an individual user.

The system was tested according to above two schemes. In writer dependent scheme 20 people whose writing samples were used in training phase were put into test. In writer independent scheme writings of 20 new users whose writing samples were not used in training phase were put to test.

After conducting the test schemes 1 and 2, it was found that some characters were frequently misclassified. Figure 6 shows the misclassified characters.

ഝ	ഞ	ണ	ള	ം	്യ	ാ	ി	ീ	ു	ൂ

Figure 6. Sample of erroneous characters

When prototype selection using Hierarchical Clustering was included in the training module the percentage of misclassification was reduced. Table 1 illustrates comparison of error rates of misclassified characters obtained while executing methods with and without prototype selection using Hierarchical Clustering. To improve the quality of recognition of such characters more samples were also included in the training set. Now the training set consisted of 2591 samples.

TABLE I. ERRONEOUS CHARACTERS AND THEIR MISCLASSIFICATION RATES

Input character	Mis-classified Character	Error rate without prototype selection	Error rate with prototype selection
ഞ	ണ	0.10%	0.06%
ണ	ഞ	0.08%	0.08%
ള	ഉ	0.40%	0.12%
ം	ഠ	0.79%	0.04%
്യ	ൃ	0.16%	0.08%
ീ	ി	0.21%	0.11%
ാ	ി	0.25%	0.05%
ൂ	ു	0.36%	0.09%

Performance of the system is summarized in Table 2. The overall performance of the earlier system [3] using the feature vector of a combination of context bitmap and normalized (x, y) coordinates was 88.75 %. In the current system the overall accuracy was found to be 98.125%. Another performance highlight of this system is that even when the sample of 1 person was used for training, an accuracy of ~92.165% was obtained. Thus this system provides higher accuracy with even a small sample size.

TABLE II. VARIOUS SCHEMES AND ITS ACCURACIES

Scheme	Total No. of training samples	Accuracy
1	2560	99.015%
2	2560	97.75%
1	2591	99.625%
2	2591	98.125%

V. CONCLUSION

The system for Online Character recognition for Malayalam developed was able to read handwritten characters and match them to canonical representations it was trained for. K-NN classifier algorithm was used to train and recognize the character.

The system exhibited an accuracy 98.125% with a recognition time of 15-30 milliseconds in a machine having the configuration of AMD Athalon 2.0 with 512MB RAM. The recognition time was found to be reduced to 0-16 milliseconds in a high performance machine having the configuration of Intel(R) Xeon(R) 5160 @ 3.00GHz with 3.00GB of RAM.

REFERENCES

[1] G.Raju "Recognition of Unconstrained Handwritten Malayalam Characters using Zeero-crossing of Wavelet Coefficients," Proc. of 14th international Conference on Advanced Computing and Communications, 2006, pp 217- 221J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] V.S Roshini, Shanifa Beevi, Revathy, "Machine Recognition of Malayalam Characters," Proceedings of the International Conference on Cognition and Recognition, 2005, pp 477-481.

[3] M.Sreeraj, Sumam Mary Idicula, "On-Line Handwritten Character Recognition using Kohonen Networks" Proceedings of the IEEE 2009 World Congress on Nature & Biologically Inspired Computing (NABIC '09),2009 pp 1425 - 1430.

[4] UNIPEN format, *http://unipen.nici.ru.nl/unipen.def.*

[5] I.Guyon, L.Schomaker, R.Plamondon, M.Liberman, S.Janet,, "UNIPEN Project of Online Data Exchange and Recognizer Benchmarks", International Conference on Pattern Recognition (ICPR 1994),Jerusalem, Israel(October 1994).

[6] Brijesh Verma, Jenny Lu, Moumita Ghosh, Ranadhir Ghosh "A Feature Extraction technique for Online Handwriting recognition," IEEE International Joint Conference on Neural Networks, 2004, Hungary, IJCNN'04, pp. 1337-43.

[7] S.Jaeger, S. Manke, J. Reichert, A. Waibel, "Online handwriting recognition: the NPen++ recognizer," International Journal on Document Analysis and Recognition, March, 2001, vol.3 (3,) pp. 169-180.

[8] R.Balaji, V.Deepu, S.Madhvanath, J.Prabhakaran, "Handwritten Gesture Recognition for Gesture Keyboard", In Tenth International Workshop on Frontiers in Handwriting Recognition (2006).

[9] M.Pastor, A. Toselli, and E.Vidal, "Writing Speed Normalization for On-Line Handwritten Text Recognition,"Proceedings of the International Conference on Document Analysis and Recognition (ICDAR), 2005.

[10] I. Guyon,P . Albrecht,Y. Le Cun,J. Denker,W. Hubbard,"Design of a Neural Network Character Recognizer for a Touch Terminal," Pattern Recognition,1991,24(2):105–119.