Asim Munir

HMM Based Online Urdu Character Recognition

Developed By

Muhammad Imran Razzak
255-FAS/MSCS/F05

Khurram Shehzad
06-FAS/PHDTE/S05

Supervised By
Dr. Syed Afaq Husain

Faculty of Basic and Applied Sciences,
Faculty of Engineering and Technology,
International Islamic University,Islamabad
2007

MS
006.42
RAH

1- Image processing - Digital techniques
2- Optical character recognition devices.

In The Name of

ALLAH ALMIGHTY

The Most Merciful, The Most Beneficent

A dissertion submitted to Faculty of Basic & Applied Sciences and Faculty Of Engineering & Technology, International Islamic University, Islamabad, Pakistan as partial fulfilment of the requirements for the award of the degree of

MS

Faculty of Basic and Applied Sciences
&
Faculty of Engineering and Technology
International Islamic University, Islamabad

Date: 05-10-2007

Final Approval

It is certified that we have read the thesis submitted by **Muhammad Imran Razzak** 255-FAS/MSCS/F05 and **Khurram Shehzad** 06-FAS/PHDTE/S05. It is our judgment that this thesis is of sufficient standard to warrant its acceptance by the International Islamic University, Islamabad for the MS Degree.

Committee:

External Examiner
Dr. M.A. Ansari
Associate Professor
Federal Urdu University Islamabad

Internal Examiner
Asim Munir
Assistant Professor.
International Islamic University.

Supervisor
Dr. Syed Afaq Husain
Professor.
Air University, Islamabad

DEDICATION

We dedicate this effort to our Nation and National Language. National Language is our recognition and professionals should step forward to promote our language in this new Era of technology.

Muhammad Imran Razzak
255-FAS/MSCS/F05
Khurram Shehzad
06-FAS/PHDTE/S05

DECLARATION

We hereby declare that this project report, neither as a whole nor as a part thereof has been copied out from any source. It is futher delared that we have developed this project and acoompanied report entirely on the basis of our personal efforts made under the sincere guidance of our teachers. No portion of the work presented in this report has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning. If any part of this report is proved to be copied out or found to be reported, we shall stand by the consequences.

Muhammad Imran Razzak
255-FAS/MSCS/F05
Khurram Shehzad
06-FAS/PHDTE/S05

ACKNOWLEDGEMENT

All praise is to Allah AlMighty, The Most Compassionate and Merciful, The Most Gracious and Beneficent, Who bestowe us good health, courage and knowledge to complete our work.

We feel very previledged to thank our supervisor Prof. Dr. Syed Afaq Hussain for their continous support and assistance. In the end we would like to thank all our class fellows and friends for helping us.

Muhammad Imran Razzak
255-FAS/MSCS/F05
Khurram Shehzad
06-FAS/PHDTE/S05

ABSTRACT

With the advent of new input devices there is a need to provide the users with natural way of input. Obviously the natural way of input text is using Pen and Paper. The new devices such as PDAs, Writing Boards are also natural ways of input. There is another device Digital Pen which writes on normal Paper. The paper is printed with predefined pattern. Pen and Paper are the most natural way of entering the handwritten data. There are number of online handwritting recognition engines available for languages such as English, French, Spanish, Chinese and Arabic. Cursive languages espically, are still a focus point for the researchers. The variation in writing style and cursive nature adds to the complexity of Recognition.

We propose a technique for On-Line Handwritting Recognition using Hidden Markov Model for Urdu with Digital Pen(Anoto technology). In this approach we separate the legature on the basis of size and identify the secondary stroke. Once the Legatures are separated we consider first stroke as primary stroke. The chain code and features are computed for the primary stroke and passed on to Hidden Markov Model for identification. Once the Primary Stroke and Secondary Stroke is identified grammer is used to identify the exact legature. We have separate Grammar for each legature with similary primary stroke.

PROJECT IN BRIEF

Project Title:	Hidden Markov Model(HMM) Based On-Line Urdu Character Recognition
Under Taken By:	Muhammad Imran Razzak Reg. No. 255-FAS/MSCS/F05 Khurram Shehzad Reg. No. 06-FAS/PHDTE/S05
Supervised By:	Dr. Syed Afaq Hussain International Islamic University, Islamabad.
Starting Date:	1 January, 2007
End Date :	31 July, 2007
Tools and Techniques	Digital Pen (Nokia SU-27W) (Anoto Technology) Matlab 7.0 Visual C# dot net Windows XP.

Table of Contents

Chapter-1
INTRODUCTION

1.1 Project Introduction:

Our research is On-Line Handwriting Recognition using Hidden Markov Model for Urdu script. The only purpose of our research is to extend the work of our senior student to a level where we can implement it practically in devices like PDAs, and Mobiles. We are sure that after couple of revision we will see Urdu recognition available in devices.

Very limited work has to be done on online side and unfortunately there is no recognition engine available in the market for both online and offline character recognintion. Whole work done on Urdu recognition is based on Artifical Neural Network and we focused on Statistical Modeling technique namely Hidden Markov Model due to its success in similar application like character recognition and speech recognition. As we have time information in online system thus it produce better reuslt for online system than offline character recognition that have no time information. Also we can compare the results of recognitions using statistical and Artificial Neural network so that it can serve as reference to new researchers. Here is a brief introduction on Online Handwritting recognition also termed as Online Optical Character Recognition.

1.2 Optical Character Recognition

Optical Character Recognition is a field of research in Artificial Intelligence, Pattern Recognition and Machine Vision. Optical Character Recognition (OCR) is the branch of pattern recognition that studies methods of converting text in the form of images into computer understandable text. This deals with extraction of Text from images or any other signal and its conversion to Computer or Machine understandable text. The main purpose of OCR is to add human reading power to our machines. The potential of OCR systems is enormous because they enable users to harness the power of computers to access printed documents. OCR is already being used widely in the legal profession, where searches that once required hours or days can now be accomplished in a few seconds.

While Optical Character Recognition has made huge advances in recent years, it still does not perform well in recognizing handwriting or fonts that look similar to

handwriting due to large variation of writing. There are systems within the banking industry that use OCR technology to try to read the amounts on hand written checks, to go along with the computer's ability to read the routing and account numbers.On the basis of input data to the Optical Character Recognition System we can classify the OCRs into two categories. Which are as under.

1.2.1 Offline Character Recognition

Input data in the Offline Recognition is mostly in the form of image, bitmap or raw document. The input data is mostly collected from scanner or image from digital camera. The technique is referred to as offline as the recognition is done after the document is complete. Means in these types of recognition systems the recognition is not performed at the time of writing and there is some delay involved. The Urdu script is very similar to the Arabic script. Arabic Text Recognition Systems generally have following stages: image acquisition,pre-processing, segmentation, feature extraction, classification and recognition[1] .

1.2.2 Online Character Recognition:

On-line handwriting recognition is a very important technique for convenient human computer interface. Pen-based input gives a lot of advantages. First of all, it helps users, such as computer novices and old people, to conveniently use a computer. It also makes a small size portable computer (PDA, handheld PC, palm PC, etc.) possible because there is no need for keyboard or keypad. The data is input through some Digital Pen, Writing Board and Styles. The device transfer the coordinates of the data to the system. For on-line recognition, a digitizer samples the handwriting to time-sequenced pixels as it is being written. Hence, the on-line handwriting signal contains additional time information, which is not presented in the off-line signal.

1.3 Classification of OCR

From the classifier perspective, the Arabic Text Recognition Systems are further divided into Segmentation based and Segmentation-free systems [2].

1.3.1 Segmentation Free Systems:

In these systems, the word is recognized as a whole without trying to segment and recognize characters or primitives [3]. There are many approaches to this method.

Keeping in mind the Urdu script especially the handwritten Urdu script is simply very difficult to segment individual characters from a legature. In ligature based approach every ligature is recognized based on some pre defined features of ligatures. This approach does not need character segmentation algorithms, but efficient ligature recognition algorithms. A few rules are needed and the ligatures from all possible Urdu words are to be stored in the data set/dictionary.

1.3.2 Segmentation Based Systems:

In Segmentation based systems, each word is further divided into a number of subparts. The segmentation-based systems are further subdivided into four categories:

- Isolated/Pre-segmented characters,
- Segmenting a word into characters,
- Segmenting a word into primitives,
- Integration of recognition and segmentation.

These systems are either impractical because they try to recognize digits and isolated characters or they have low recognition rate because of segmentation errors [3].

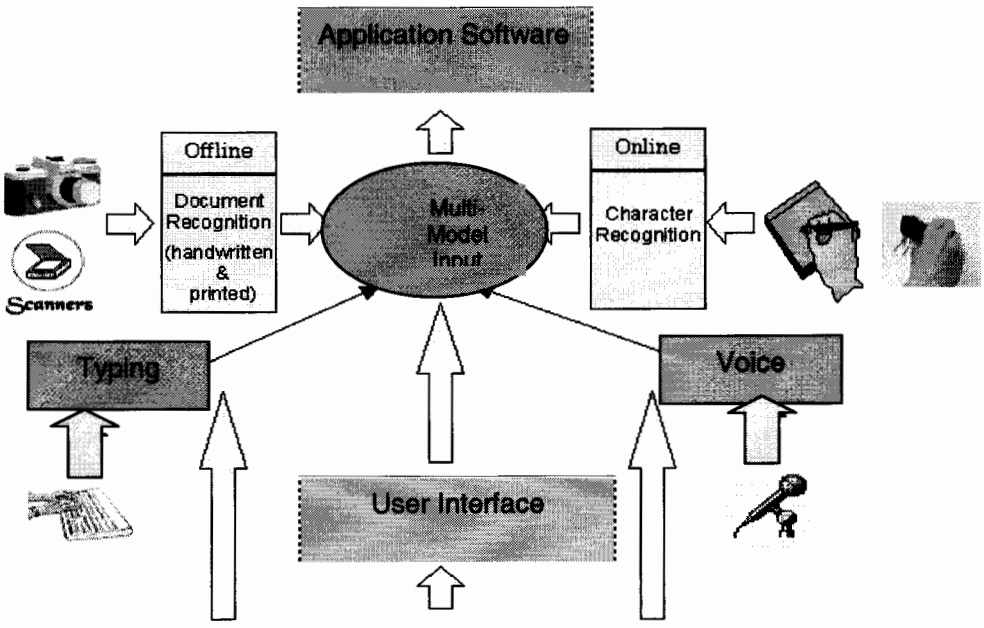

Figure 1.1 Online and offline OCR

1.4 Methods for Recognition:

The Process of recognition starts with data acquisition, preprocessing, segmentation, Feature Extraction, Recognition and than post processing.

1.4.1 Data Acquisition:

The data is collected from the device. The data we receive from the device contains a stream of {x,y} coordinates. The time information of the stroke. Means we can get the amount of delay involved in between the strokes. Moreover there are devices that provides the level of force with which user has pressed the pen. Data is acquired once the pen is up or it can be acquired on the fly in realtime.

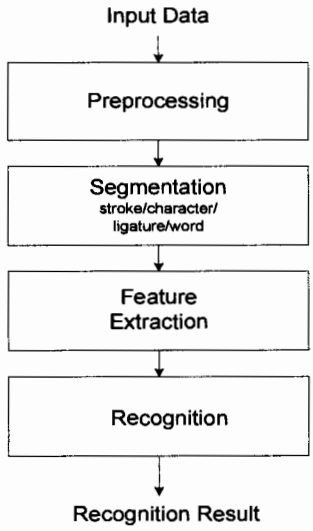

Figure 1.2: Recognition Process

1.4.2 Pre-processing:

Preprocessing invloves the removal of any redundant information and noise. This process is a collection of different operation that are performed over a raw data. This raw data can be image or stream of x and y. Following are different steps involved in the Pre-processing.

- Skew Detection
- Noise Removal
- Filtering

- Smoothing
- Thinning
- Resizing
- Normalization
- Slant Normalization
- Decomposition.

1.4.3 Segmentation:

The segmentation process takes all of the written data and attempts to segment this data into words and characters. This process also incorporates global features such as baseline, size, and other helpful statistical features above the shape-based recognition. All of these features are combined together and use optimization techniques, which output the most probable segmented recognition results in a short time.

1.4.4 Feature extraction:

The raw data (x, y coordinates) is transformed into more suitable recognition related features. These features model the underlying features of the written data, such as the curve, direction, break points, height and more. These features are the groundwork for the higher levels.

1.4.5 Shape Recognition/ Post-Processing:

The heart of online/ offline character recognition is the ability to compare a written set of strokes (or sub strokes in cursive letters) to character templates. The results are a set of characters along with their associated match probability. The comparison is based on analyzing the shape features, with the more sophisticated (high level) attributes assisting to provide a complete recognition system later.

1.4.6 Linguistics and Dictionary:

These are additional sources of information that help to resolve conflicts between similar looking characters. The information is usually based on statistical modeling of the language or as a language dictionary. The statistical representation optimizes the written text as an adequate sequence of letters, as expected in the language – such as referring to "ing" at the end of a word, versus "iny". The dictionary searches the written text for the most probable word in the dictionary.

1.4.7 Training:

Training enables the user to teach the recognition system his/her individual writing style. Preferably, the training is done in an "on the fly" manner – i.e. any correction of erroneous text is also a training event.

1.5 Urdu Character Recognition

Urdu is a cursive language. For designing a recognition system for any language it is necessary to understan the rules and characteristics of that language. So here is the brief history of urdu

1.5.1 History Of Urdu:

The word "Urdu" is derived from Turkish language meaning 'foreign' or 'horde'. It belongs to the Indo-European language family and has influences from Persian, Arabic and Hindi. More than 160 million people use Urdu as its first and second language across the globe [2]. Urdu is the national language of Pakistan and is the mother language of about 10 million (8%) people in Pakistan and 50 million in India. Urdu is also widely spoken in Afghanistan, Bahrain, Bangladesh, Botswana, Fiji, Germany, Guyana, India, Iran, Malawi, Mauritius, Nepal, Norway, Oman, Qatar, Saudi Arabia, South Africa, Thailand, UAE, United Kingdom and Zambia. Formal vocabulary of Urdu is borrowed from Arabic and Persian while it embodies morphological and phonological similarities with Hindi [3].

1.5.2 Urdu Characterestrics

Urdu contain 36 characters for which Urdu script uses different style. More than ten fonts are used in Urdu namely Nastaliq, Naskh , Noori Naskh, Noori Nastaliq , Koofi ect Nastaleeq and Nasakh are the most popular.Some of the most important characteristics of Urdu script , which distinguish it from other languages, are:

- **Direction of Writing:**

Unlike English, Urdu is written from right to left like Arabic.

- **Cursive Nature:**

Urdu text, both typewritten and handwritten is cursive in nature with all the characters connected to each other within a word. Spaces in Urdu may occur between ligatures and between words.

• **Presence of a Base Line**:

Like other languages e.g. English, Urdu has a base line. The base line is a horizontal line which runs through the text, cutting all the words at some point.

• **Overlapping:**

The characters in Urdu overlap vertically and do not touch each other.

• **Diacritics:** Urdu text has some diacritics. Diacritics though sparingly used, are very important in the proper pronunciation of the word. The examples of some of the diacritics are: Kasr-e-Izafat, jazm, hamza, khari zabar, and khari zer etc. [3].

Ligatures: Several characters of Urdu are combined vertically to form a ligature.

• **Upper and Lower Cases:**

There is no concept of upper case and lower case in Urdu language writing.

• **Context Sensitive Shape**

The shape of every character depends on the position in the word. In the Naskh way of writing script, Urdu assumes four different shapes depending on whether the character is isolated, in the beginning, at the end or connected from both the sides in a word as shown in figure 1.

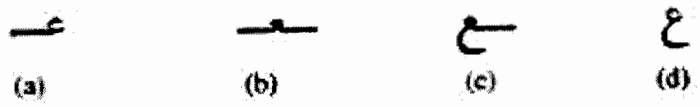

Figure 1: Different shapes of Urdu character (a). beginning (b). middle. (c) end.(d)Isolated [1]

• **Strokes:**

The basic rule is that any Urdu character has one main stroke and zero or more secondary stroke as shown in figure 2. Each word of Urdu can be broken into ligatures. The combination of these ligatures forms different words. Each ligature comprises of more than one strokes.

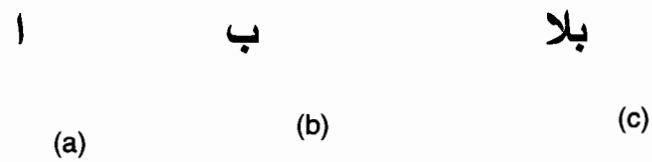

Figure 2. (a). zero secondary stroke,(b). One secondary stroke

(c). Two Secondary Stroke ie dot and digonal of lam

1.5.3 Applications of Online Urdu Character Recognition:

On-line handwriting recognition is a very important technique for convenient human computer interface. Pen-based input has a number of advantages. First of all, it helps users such as computer novices, old people and house wives to conveniently use a computer. It also makes a small sized portable computer (PDA, handheld PC, palm PC, etc.) possible because there is no need for keyboard or keypad. The collected raw data is later used for the recognition process.

The specific applications of this new software for Urdu are expected in areas where natural way of input i.e. pen-based input is preferred as the convenient input methods:

- Mobile phones giving the facility of Urdu messaging,
- Urdu digitizing notepads,
- Lecture to text conversion,
- User friendly Urdu handwriting recognition software such as Microsoft Office.
- Entering Urdu text in to PDA's.

1.6 Organization of Thesis

There are five chapters and one appendix in the thesis/research report. The first chapter introduces Optical Character Recognition (OCR), its types and different classification methods. A brief introduction on Urdu and its OCR is also given in this chapter. The second chapter gives brief overview of some previous work done in the field of OCR. The third chapter presents our proposed methodology. It describes the working of each module of the system. The forth chapter describes the testing of different samples. The fifth chapter gives brief discussion of the conclusion and concludes the discussion with future recommendations.

Chapter-2
LITERATURE REVIEW

This chapter discuss some of the previous works done in the field of Optical Character Recongition. Following are the some research papers, which have been reviewed in order to understand what has been done in the past.

2.1 Automatic Recognition of Handwritten Arabic Characters Using Their Geometrical Features [4]

This research aims to use geometrical features and neural networks to automatically recognize (read) off-line handwritten Arabic words. It concentrates on the feature extraction process, i.e. extraction of the main geometrical features of each of the extracted handwritten Arabic characters. A complete system able to recognize Arabic-handwritten characters of only a single writer is proposed and discussed. A review of some of the previous trials in the field of off-line handwritten Arabic character recognition is included. The system first attempts to remove some of the variations found in the images that do not affect the identity of the handwritten word (slant correction, slope correction, and baseline estimation). Next, the system codes the skeleton of the word so that feature information about the lines in the skeleton is extracted (segmentation and feature extraction). The features include locating endpoints, junctions, turning points, loops, generating frames (segmentation step), and detecting strokes. These features are then passed on to the recognition system for recognition. The character classification is achieved in this research using a feed forward error back propagation neural network. A 69.7 percent recognition rate has been achieved for the character frames of data.

Preprocessing:

The preprocessing operation consists of image Loading, Slope Correction, Slant Correction, and Thinning. The slope correction is achieved by application of the Shear transform parallel to the y-axis.

Finding Handwriting Features:

A number of useful features have been found from the processing that has already been performed on the writing: endpoints, junctions, complementary characters, loops, and turning points. The methods for the detection of intersection points, endpoints, and loops, are all operating on skeletonized bit maps.

Character Classification:

The character classification is done in this research using feed forward error back propagation neural network. The network has a single hidden layer of standard perceptions with nonlinear activation functions. The mapping process is from input, represented by features extracted for the Arabic character, to the output, that represents an indication to that character.

The recognition process has known two trials. The neural network has three layers. In both trials, the number of output neurons and the number of hidden neurons was same. The achieved accuracy in first trial is 53%. In the second trial, the recognition accuracy increases to 69.7%.

2.2 A Feature Extraction Technique for Online Handwriting Recognition [5]

The paper presents a feature extraction technique for online handwriting recognition. The technique incorporates many characteristics of handwritten characters based on structural, directional and zoning information and combines them to create a single global feature vector. The technique is independent to character size and it can extract features from the raw data without resizing.

Their proposed technique has been classified into eight modules such as dehooking, extract feature points, stroke extracting, calculate PEN-UP, extract zones and directions of start point and end point, extract changes in writing direction, calculate height/width ratio and extract zone information which creates a global feature vector and uses a back-propagation neural network based classifier. The models are described below:

Dehooking:

Hooks can occur at the beginning and end of strokes due to inaccuracies in pen-down detection and rapid or erratic motion in placing the stylus on, or lifting it off the tablet. To remove hooks if the vector direction length is less than threshold, remove it from the dataset, otherwise keep it.

Extract stroke:

Stroke is defined as continuous path of the pen from the moment it is placed on the writing surface until the moment it is lifted up. In this case, stroke is the series of

points from "PEN-DOWN" point to "PEN-UP" point. The feature calculated in this research is the number of strokes for one character. Thus the method used to get the strokes was simply by counting how many "PEN_DOWN" occurred in the dataset for one character or digit.

Zones and directions of start and end point:

Using the lateral coordinates and the longitudinal coordinates of the first point and the second point, start point direction (SD) and the end point direction (ED) has been calculated. For the zone information, the whole region of character was separated into six zones in this research.

Change of writing direction:

The change of writing direction is regarded as the changing from pen going up (down) to down (up) or going left (right) to right (left). For one particular character or digit, the order of strokes may be very different, but the change of writing direction will be similar. Based on vector direction, we can get the jag point where the writing direction changed. Using the coordinates of continuous two jag points they have found how many times the direction is changed.

Global Feature Vector:

A global feature vector is based on a number of characteristics such as writing direction going down, writing direction going up, writing direction going left, writing direction going right, Z1, Z2, Z3, Z4, Z5 and Z6: 6 zones.

Back-Propagation Neural Network Classifier:

A back-propagation neural network with a single hidden layer is used as a classifier.Using the proposed technique and a Neural Network based classifier; many experiments were conducted on UNIPEN benchmark database. The recognition rates are 98.2% for digits, 91.2% for uppercase and 91.4% for lowercase. The method and techniques proposed in this paper have shown improvement when compared with previously existing techniques.

2.3 Design of an OCR System for Online Urdu Input[6]

The recognition engine Online Urdu Character Recognizer (OLUCR) presented by Hussain S.A.et.al recognizes 38 one character ligatures , 709 two character ligatures e.g. شا, جب etc and approximately 50 most commonly used three character ligatures

e.g. تھا, ہیں, کیو, میں, لیا etc. The authors presented a method for recognition of online Cursive Urdu hand written Nastaliq Script. The system is currently trained for 250 ligatures. By using multiple classes of features, they improved the number of ligatures that can be identified and successfully recognized 250 base ligatures and 6 secondary strokes. These when combined form more than 850 ligatures which can recognize approx 18000 words of Urdu dictionary successfully. The Recognition rate of base ligatures was 93% and of the secondary strokes was 98%.

2.4 On-Line Recognition of Handwritten Text Based on Hidden Markov Model [7].

New use of pen pressure as a feature was proposed for the improvement of basic performance of the writer independent online handwritten character recognition technique based on HMM technique.

He proposed two kind of feature related to pen pressure one is the pressure representing the pen ups and downs in a continuous manner and other is the time derivative of the pressure representing the temporal pattern of pen pressure. Through experimentation evaluation using 1016 elementary Kanji characters compared with the base line performance using the velocity vector only.

2.5 Recognition of Online Handwritten Formulas [8]

The paper presented by Kosmala A., Rigoll G. describes a system for the recognition of online handwritten mathematical expressions. The system presented here is based on Hidden Markov Models (HMMs) and offers thus the advantage of a simultaneous segmentation and recognition, avoiding the complex and crucial handwriting segmentation during pre-processing. The segmentation and recognition result is used in a further step for the interpretation of the symbols and their spatial relationships The results presented in this paper show, that with the introduction of appropriate constraints a robust online handwritten formula recognition is feasible, even without requiring an artificial style of handwriting. Furthermore it is very helpful to perform segmentation and recognition in a single path in order to achieve a robust recognition, instead of applying segmentation algorithms in a pre-processing step.

2.6 On-Line Handwriting Recognition Using Hidden Markov Models [9]

In this thesis Han Sue presented a series of feature experiments aimed at showing that the performance of the baseline system can be improved dramatically by augmenting the six baseline features with new features, which would provide the HMM with information about the handwriting which was not represented by the original features. A new vertical height feature was used to characterize vertical height. A new space feature was used to represent inter-word space. The hat stroke features were used to overcome HMM's output independence assumption. The sub stroke features were implemented to improve the characterization of global information. By training and testing on the BBN on-line cursive handwriting data corpus, with the new features the system obtained a word error rate of 9.1%, a 34% reduction in error from the baseline error rate of 13.8%. The space feature and the sub stroke features each reduced the word error rate approximately 15%. The new features improved the HMM's modeling of handwriting, thus, also improved the recognition performance of the overall system significantly.

2.7 Online Handwriting Recognition Technology and Its Applications [10]

The study presented by Tanaka H, Iwayama N. describes Fujitsu's online handwritten character recognition (OLCR) technology and some application software that adopts this technology. OLCR has the highest level of performance among Japanese OLCRs and is based on two of unique character recognition technologies: hybrid character recognition and bigram-based context processing. To realize more effective and practical handwriting interfaces, additional OLCR techniques such as hybrid adaptation, predictive handwriting recognition, and box-free handwritten string recognition were developed. Several software products, including Japanist 2003 and Japanist for Pocket PC, have adopted this technology. This technology is also used by the standard handwriting recognition engine in FMV-STYLISTIC, which is one of Fujitsu's Tablet PCs. In one experiment, OLCR technology achieved a 94.6% recognition accuracy for Japanese text compared to other software available on the market, which achieved an accuracy of only 82 to 88%.

2.8 Sub Stroke Approach to HMM-based On-line Kanji Handwriting Recognition [11]

A new method is proposed for on-line handwriting recognition of Kanji characters. The method employs sub stroke HMMs as minimum units to constitute Japanese Kanji characters and utilizes the direction of pen motion. The main motivation is to fully utilize the continuous speech recognition algorithm by relating sentence speech to Kanji character, phonemes to sub strokes, and grammar to Kanji structure. The proposed system consists input feature analysis, sub stroke HMMs, a character structure dictionary and a decoder. The present approach has the following advantages over the conventional methods that employ whole character HMMs.

2.9 An Online Handwriting Recognition System For Turkish [12].

They presented an online handwriting recognition system for Turkish. The results are quite good and show the promise of the features and the overall approach of the system. Sample handwritings from the database are shown in Figure 6. Most words were written discretely (noncursive), with occasional touching characters, which, unlike offline handwriting, do not pose extra difficulty for the recognizer. However, on close inspection of the errors made by the system, while writing a word, many people have gone back and rewritten a bad character somewhere in the word. Since word models are composed of concatenated letter models of its constituent letters, in order, this writing behavior may account for a significant part of the error rate.

Chapter 3
PROPOSED SYSTEM

3.1 Problem Statement:

There are many online character recognition systems for different languages for example English, Japanese, Chinese and Thai etc. Urdu has been unfortunate in this respect. There is not a single online character recognition exist for Urdu. Urdu is a very complex language when it comes to recognition by computer. It is difficult to recognize because of having quite distinct and complex characteristics such as

- Variety of scripts and styles.
- Cursive script.
- Vast character set.
- Context Sensitive letter shape.
- Overlapping characters.

3.2 System Design:

The recognition engine makes use of various approaches in order to recognize the strokes. This is due to the cursive nature of the Urdu handwriting. The characters are recognized using the ligature based approach in which the whole ligature is recognized as it without segmenting it into its constituent alphabets. The recognition systems are generally divided in to two types. Segmentation based and segmentation free recognition systems. We have used the segmentation free approach because many of the recognition errors occur due to errors in segmentation. The segmentation free system extracts a feature vector for each ligature which is then passed on to HMM for classification of the ligature. We have used the Vitribi algorithms for classification. Using the strokes (x, y) co-ordinates and the chain codes, unique features for every stroke are detected and a feature vector is extracted. This feature vector is then fed in to the back propagation neural network for the classification of every stroke in to its respective class.

We have successfully recognized 700 base ligatures and 8 secondary strokes. The secondary strokes recognized are:

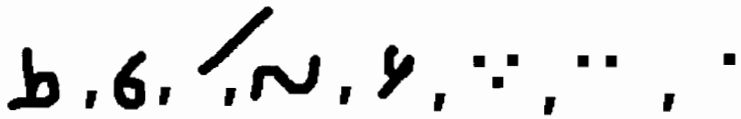

Figure 3.1: Secondary Strokes

Namely, (left-right order) hay, kaaf and gaaf long diagonal stroke, Madaa, Hamzaa, triple dots, double dots and the single dot. Altogether the recognizer can successfully recognize ligatures whose combinations results in the formation of words.

The valid ligatures form valid words. These words are then written to a text file for display and further editing as per user requirement.

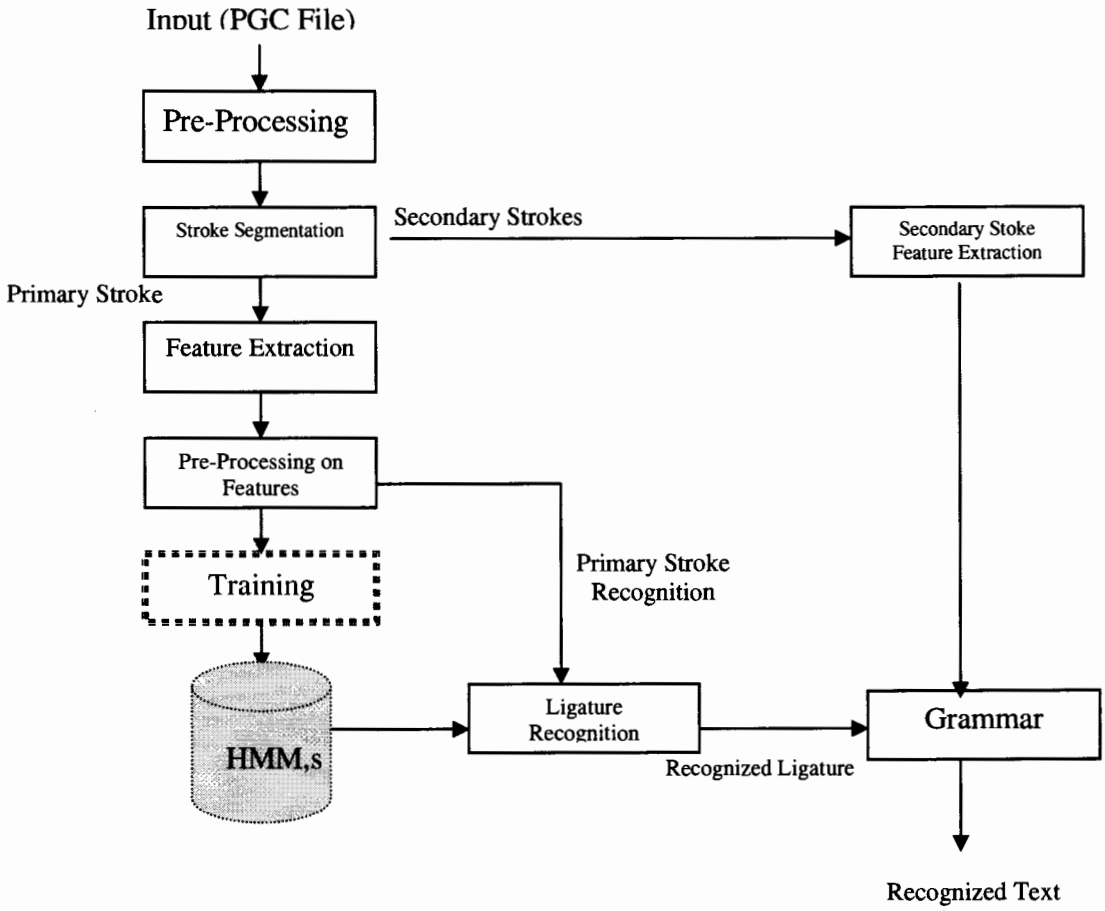

Figure 3.1 Structure of the System

3.3 System Modules:

Keeping the challenges of Urdu online character recognition in mind and following the literature study, our proposed system consists of the modules which are also the building blocks in many of the papers mentioned in the Literature Survey. Methodology used for our system is online character recognition. As recognition can be either word, ligature or character based. Our system uses ligature based recognizer, thus this is the segmentation free approach. The modules include:

3.3.1 Acquisition:

In this system digital pen was used for ease of use. As pen have limiting processing power and camera frame rate, therefore input was taken at a normal speed from the trained user.

3.3.2 Preprocessing:

The data thus obtained often contains irregularity such as the different size, hooks and erratic handwriting generated by inexperienced users. Hooks occur due to the inaccuracies during pen up and pen down.

Figure 3.2: Jeem written by inexperienced writer and Lam Ray containing hooks in the beginning and end

3.3.2.1 Interpolation:

Due to the limiting processing power of pen and low camera frame rate, it skips some points; this depends upon the writing speed, to compute the missing point's interpolation is performed for correct classification. More over for better result the writer should write with normal speed. Some important feature missed when high speed text was written by user for training purpose.

3.3.2.2 Scaling:

As the user writes text with different size, so it must be resized before feature extraction so that variations are reduced.

3.3.2.3 Chain Codes:

Chain code describes an object by a sequence of line segments with a given orientation. The method was introduced in 1961 by Freeman. In this approach, an arbitrary curve is represented by a sequence of small vectors of unit length and a limited set of possible directions, thus termed the unit-vector method. From a selected starting point, a chain code can be generated by using 4-directional or 8-directional chain code. N-directional ($N = 2k$) chain code is also possible, it is called general chain code. The codes associated with eight possible directions, with x as the current contour pixel position, are defined as:

$$\textit{Chain codes} = \begin{matrix} 3 & 2 & 1 \\ 4 & x & 0 \\ 5 & 6 & 7 \end{matrix}$$

We have used the 8-directional chain coding. The pre-processing steps were performed on the chain codes.

In order to remove hooks and the erratic strokes the following approaches were carried:-

1. Dehooking.
2. Smoothing.

3.3.2.4 Dehooking:

Hooks are very common artifacts found at the beginning and end of the strokes. As the pen is very sensitive thus they are generated during fast writing or writing by inexperienced person, when pen-down and pen-up events are generated. These often create problems in the detection of the original ligature. Therefore, it is very important to remove them.

These usually occur at the beginning and the end of the stroke. The hooks occurring at the beginning and the end of the stroke are removed with the help of the

generated chain codes. If the length of the chain code at the beginning or end is less then the specified threshold, then that part considered a hook and is removed by either discarded it or replacing the respective co-ordinates with the neighboring ones.

Hooks are generated by users either he is experienced or inexperienced. As there is a very small lines that is added at start and end but some problems exist in removing these hooks, it may possible that small up of Jeem is removed as shown in below instead of hooks.

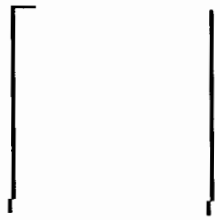

Figure 3.3: Alif before and after Dehooking

3.3.2.5 Smoothing:

As the text contains the irregularity due to the hand shivering by nature. Smoothing is one of simplest approaches for data filtering. As in many preprocessing methods, it consists of substituting the coordinates of the original point by using the neighboring points. In our project smoothing was done on the chain codes of the stroke. Two to three pixel smoothing was done as per the variations in the chain codes of the stroke for e.g. the Urdu letter Laam gave the following chain codes before and after smoothing:

Before:

"6, 6, 5, 5, 5, 5, 5, 5, 6, 5, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 7, 7, 6, 6, 6, 6, 6, 6, 6, 6, 6, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 4, 5, 4, 4, 4, 4, 4, 4, 4, 4, 3, 2, 3, 2, 3, 3, 3, 3, 2, 3, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 1, 2, 1, 2, 1, 2, 1, 2, 2, 1, 2"

After:

" 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 3, 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2"

To grasp the idea better the e.g. operation is shown in the figure below.

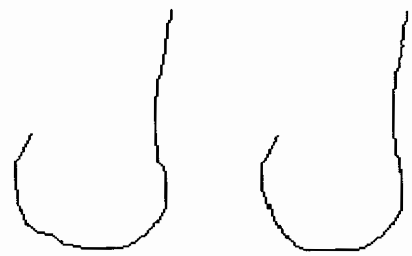

Figure 3.4: Laam before and after smoothing

3.3.3 Ligature Identification:

Once the ligature has undergone the two steps of pre-processing it is ready for identification. For identification of the ligature a segmentation free approach is used. In this approach the ligature is not broken up in to its constituent alphabets and recognized by constructing a feature vector of every ligature that is input to the system. Although, this process is somewhat difficult, as we had to find out unique features for every ligature out of the hundreds of ligatures under our study. With The Grace of Allah Almighty we managed to find out a unique feature set for each of the **500** ligatures successfully.

3.3.4 Feature Extraction:

In this stage, we extracted features for the recognition of base ligatures and the secondary strokes as well. For the base ligatures a feature vector consisting of twenty six was prepared. The features extracted were syntactical i.e. they identified various shape forms present in the Urdu ligatures such as loops, intersections, loops in the beginning or end or the pen movement, direction/writing style of any ligature. These also included

features that are selected on the presence of certain alphabets of Urdu language. For example there is an Aein feature which is selected on the presence of Aein in any ligature. These features were very helpful in uniquely distinguishing the ligatures. The feature vector prepared includes the following features.

- **Start-Eight Chain Code:**
  This feature depends upon the starting of the ligature either diagonal, left, up act like اب.
- **Start Vertical Down:**
  This feature was selected when the ligature was a straight vertical downward in the beginning. For e.g.ط،ل،١.
- **Start Vertical Up:**
  This feature was selected when the ligature was a straight vertical upward in the beginning. As there is no word which starts from upward but this feature is used to differentiate numerals like ١ and ۱ are written with same style. So for numerals we write from downward.
- **End Vertical Up:**
  This feature was selected when the ligature was a straight vertical upward in the end. For e.g.با، کا طا.
- **End Vertical down:**
  This feature was selected when the ligature was a straight vertical downward in the end. For e.g. م, بم .
- **Horizontal RtoL:**
  This feature was selected if the during writing the ligature the pen movement is from right to left horizontally For e.g. in ب، ف.
- **Diagonal RtoL:**
  This feature was selected if the during writing the ligature the pen movement is from right to left diagonally like in ک.
- **Diagonal LtoR:**
  This feature was selected if the during writing the ligature the pen movement is from left to right diagonally like in لا.

- **Horizontal LtoR:**

  If the during writing the ligature the pen movement is from Left to right horizontally then the Horizontal lefttoright is selected. For e.g. in ے، سے .

- **Hedge RtoL:**

  In Urdu characters like Noon, Seen, Qaf have we can say semi circle sort of shape in them. For such characters we have selected a feature called the hedge. This feature is selected when semi circle present from right to left like ق، ن.

- **Hedge LtoR:**

  In Urdu characters like Jeem, Ayen have we can say semi circle sort of shape in them. For such characters we have selected a feature called the hedge. This feature is selected when semi circle present from right to left like ج,ع .

- **Curve LtoR:**

  The direction of writing of these curves varies from right to left and also from right to left. Therefore, Curve LtoR has been selected for characters those writing direction is right to left likeن ,ر .

- **Curve RtoL:**

  If the curve direction of the character from left to right then Curve RtoL is selected like سے, ج, ع .

- **Cusp:**

  A cusp is a sharp turning point in a stroke. This feature is selected for the ligature which contains the cusps such as those present in Seen and Seen Ray like سر,س .

- **Intersection:**

  When ever an intersection is encountered in a stroke this feature is selected for that stroke. For e.g. these are present in Seen, Tuwn etc. like ط ,فل .

- **Ray/Dal:**

  This feature is selected for the character ray of Urdu alphabet. If any ligature is a combination of ray or dal then this feature is also selected for that particular ligature like ر،بد،د .

- **Loop Up:**

  In order to differentiate the loop in fay, Qaf this feature was identified and selected. The writing direction of the loop in Qaf is clockwise so we selected Loop up

Feature like ف,ق .

- **Loop Down:**

  In order to differentiate the loop in fay, Qaf and Meem, this feature was identified and selected for Meem. The writing direction of the loop in Meem is anti clock wise shown in figure below. Therefore, this loop down was selected for meem like بم،جم تم.

- **Loop Swad:**

  In order to differentiate the loop in fay, Qaf , Meem and Swad, this feature was identified and selected for Swad. As the swad loop is like a egg so it is identified to separate the Swad from other loop like بص,ص .

- **Hey:**

  In order to differentiate the loop in fay, Qaf and Meem and hey , this feature was identified and selected for Hey like ہ -

- **BayRay:**

  This feature was selected if character some character combine with ray like جر , بر .

- **Bayyee:**

  This feature was selected if character some character combine with choti ye like جی, بی .

- **Aien Bit:**

  This feature was selected if character Aein is detected in any ligature like ع .

  Hay middle Bit

  This feature is selected in the presence of hay in the ligature like جہل, سہل .

  Tuan Bit:

  This feature is selected on the presence of tuan in any ligature like بطا, سطر,ط .

- **Madaa:**

  All the Madaa ligatures have this feature selected for them. The shape of madaa and other madaa ligatures for example جہ , بہ .

3.3.5 Stroke Identification:

This stage can be further sub divide in to two identification phases.

1. The identification of the base stroke

2. The identification of the secondary stroke.

In normal writing norms, the base ligature is written before the secondary stroke. Therefore, the base stroke is recognized first. This also eliminates the chance of errors such as putting a secondary stroke after a base ligature which does not expect it to be there. If we expect a secondary stroke after the base stroke, a secondary flag is set for it. Then the next stroke is treated as a secondary stroke. Another scenario is that if the user did not place a secondary stroke as it was optional. In this case the software checks the incoming stroke. If it fulfills the criteria for being the secondary stroke the processing for the secondary stroke starts otherwise this new stroke is also taken as a base stroke. For each stroke the feature vector is prepared. This feature vector is given as input to Hidden Markov Models. HMM then processes the inputs and generates the output.

3.3.6 HMM Classifier:

A Hidden Markov Model is a stochastic process that generates an output sequence, o. The output values are typically values of cepstrum parameters.

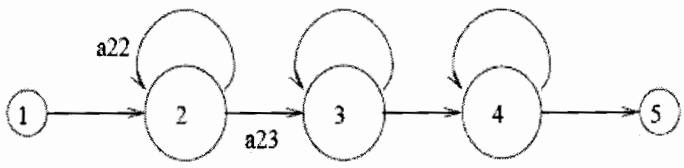

Figure 3.5: Scheme of a hidden Markov model, with circles representing states 1 to 5. The arrows represent the transition probabilities, named after the states involved. The bigger circles in the middle are the emitting states.

A difficult problem of HMM systems is to determine a method to adjust the model parameters so to fit the statistical properties of the observed data o. There is no known way to analytically find for the model a parameter set that maximizes the probability of the observation sequence. One can, however, choose the model parameters such that the likelihood, P(w|o), is locally maximized using an iterative procedure. P(w|o) is the probability of the input sequence, w, when the output sequence, o, is already known. The method commonly used for doing this, is the Baum-Welch Method.The

procedure of optimizing parameters is generally called training. To train a set of models, a set of speech data with corresponding transcriptions is used. This set, the training material, is not used when testing the performance of the speech model. Repeated training iterations is recommended to optimize performance.

The output generating process is modeled as a chain of N states, numbered 1 to N. Generally each state, except the first and last, generates output values and are called emitting states. These values are generated with certain probabilities, often modeled with Gaussian distributions, depending on the state. Between each state there are transition probabilities, i.e, the probabilities for the stochastic process to advance to the next state or stay in the current one. Each time the process visits an emitting state, a vector of output values is added to the output sequence and finally, as the process reaches the last state it is terminated. The order of states visited during a process is called a state sequence or a path. Each time a state is entered or re-entered, all transition and emitting probabilities collected during the path are multiplied to form the likelihood of the output sequence that has been created so far.

The task for an HMM-based character recognizer is to find the HMM that is most likely to have generated an output sequence, o, equal to the incoming speech sequence. The phone with the belonging HMM that has performed the best value of likelihood is then the one chosen in the recognition process.

3.3.7 Ghost Character and Secondary Stroke:

As first the ghost character is recognized and then secondary stroke are recognized then both output helps us to classify the ligature. If the primary stroke is valid then grammar is used to combine secondary stroke with the ghost character to form the ligature.

3.3.8 Output:

The output is the Urdu Text in the interface's text area and in a word file. Only the valid ligature are written in the word file. The Unicode of every ligature verified is stored and once the word formed with the ligature is identified as valid it is written to a text file.

Chapter-4
EXPERIMENTAL RESULTS

4.1 Introduction:

This is an opening step in the field of Online Urdu Character Recognition. This work was initiated with the spirit of contributing to the national language development, so that the national language can be modernized.

As Urdu is a cursive language so segmentation free approach was adopted because it is near to impossible to segmentation handwriting Urdu text due to variety of writing style. No particular script was chosen, scripts are only useful in Urdu typography or offline recognition. The online Urdu handwriting is quite different from these scripts. So we used the segmentation free approach and divided the system into two parts. First segment the primary and secondary stroke after separating them and then both output are used as an input for grammar to form a correct ligature. For this purpose, the handwriting of two writers was taken in to consideration. Along with this the most commonly used writing direction and characteristics of all ligatures under our study were taken in to account in order to generalize the system.

4.2 Selection of Ligatures:

The second important task was the selection of ligatures as Urdu has more than 20000 ligatures. As Urdu is cursive language. Being cursive implies that individual characters are combined to form words/ligatures. Therefore, we decided to start with the one, two three and four ligature words.

It is also a problem to find out the most frequent ligatures. The list of ligatures was obtained from Ahmed Mirza Jamils computerized book for inpage of more than 20000 ligatures. To find the most frequent ligature help was taken from printed median a source of Urdu language. The sources were Urdu newspapers and library of national language authority. As a result approximately six hundred ligatures including those ligatures that vary with different secondary strokes and their different positions were chosen for recognition. These six hundred ligatures form more than 1500 words which was quite an enough target for our preliminary research. A list of the ligatures is provided in the Appendix 2.

4.3 Ligature Testing and Results:

As in this system ligature base approach was used so we first recognized the ghost character, secondary stroke respectively and then combine both outputs to form the correct ligature as an input to grammar.

No	Ligature	Total Samples	Recognition Rate w.r.t. samples in %
1.	ب	10	96.5
2.	س	10	81.5
3.	ص	10	81.5
4.	ط	10	74.8
5.	با	18	96.8
6.	طا	18	71.3
7.	طب	18	83
8.	فب	18	86.3
9.	صح	16	71
10.	حح	16	82.5
11.	حد	16	97.5
12.	کص	16	93.5
13.	فک	16	87.5
14.	مک	16	87.5
15.	مع	16	75
16.	عبا	16	87.5
17.	صو	16	75
18.	طو	16	87.5
19.	سمر	16	84
20.	سے	16	80
21.	میں	16	87.5

All the ligatures were tested for correct classification in future. If a ligature was producing the bad result then it was re-sampled and train that ligature again so that proper results can be achieved. These ligatures were tested by trained user. Some of the problems encountered during this process were the difficulty of writing. As pen skips a many point and we perform interpolation but it also fails if the user writes with high speed so it may chance that some important feature will be missed. Varying speeds of writing also produced unnecessary co-ordinates. Test results of some of the difficult ligatures are given in the table below.

4.4 Secondary Strokes Identification Results:

As secondary stroke are short stroke, therefore we first segment the secondary stroke and then identify these strokes. The recognition rate of these ligatures was very good.

No	Ligatures	Samples	Results %
1		10	100
2		10	96
3		10	84
4		10	92
5		10	81
6		10	97
7		10	86

4.5 Experimental Results:

The following are the some example of testing and their results.

4.5.1 Recognition Process of Ligatures ٹ،ٹ،ڈ،ڈ.

4.5.1.1 Input:

The input was taken from digital pen as shown in figure 4.1 of character ٹ،ٹ،ڈ،ڈ.

Figure 4.1: Input character via pen

4.5.1.2 Preprocessing:

Several preprocessing operation are performed on the input text so that some drawbacks can be removed, and correct data can be inputted for classification. The following are the preprocessing performed on the ligatures ڈ،ڑ،ۃ،ۓ.

4.5.1.2.1 Interpolation and Resizing

Due to the low processing and camera frame rate pen missed some points to interpolate these point interpolation is performed. As this is handwritten system, therefore written ligature have different size, to reduce variation resizing is performed.

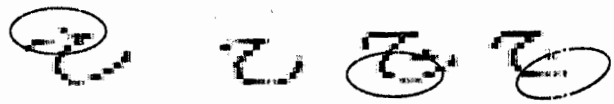

Figure 4.2: Missing data before interpolation

Figure 4.3: After interpolation and resizing of Figure 4.2

4.5.1.2.2 Segmentation of Ligature and strokes:

The primary and secondary ligatures segmented after interpolation, in the following each primary stroke with its respective secondary strokes, then segmented the primary and secondary stroke.

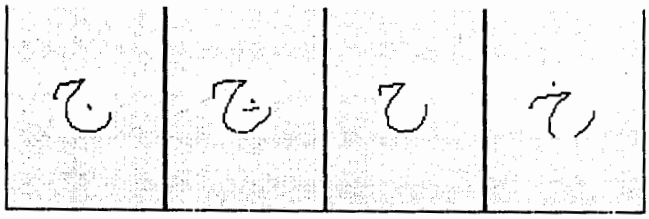

Figure 4.4: Segmentation of Primary Stroke

No	Ligature	Secondary Stroke
1	ج	One Secondary Stroke
2	ڃ	Two Secondary Strokes
3	ح	No Secondary Stroke
4	څ	One Secondary Stroke

Table 4.3: Segmentation of Secondary Strokes

4.5.1.3 Feature Extraction:

Preprocessing is also performed on the feature matrix to remove some wrong selected features. The following is the feature matrix selected for ligatures ڃ،څ،ڃ،ج after preprocessing on feature matrix.

No	Ligature	Features Matrix
1	ج	2,1,22,28,2
2	ڃ	3,1,22,28,2
3	ح	3,1,22,28,2
4	څ	2,1,22,28,2

Table 4.4: Feature Matrix

The first two features are at the starting calculated from chain code, 22 is selected when jeem features present and 28 means right hedge while ending feature is same like first two features.

4.5.1.4 Classification of Primary Stroke and Secondary Stroke.

The above feature matrix was fed to the HMM for classification. Input PGC file have four ligature, all are recognized correctly. The secondary stroke for each ligature are recognized separately.

4.5.1.5 Formation of Ligature by using Grammar:

Both output combined through grammar to from correct ligature. Grammar is given bellow.

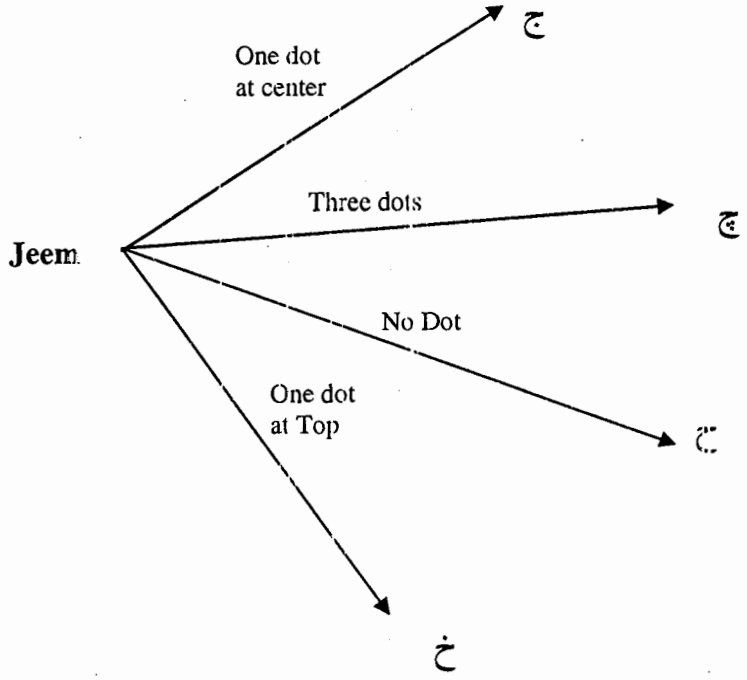

Figure 4.6: Grammar of Ligature ڃ

4.5.1.6 Output Result:

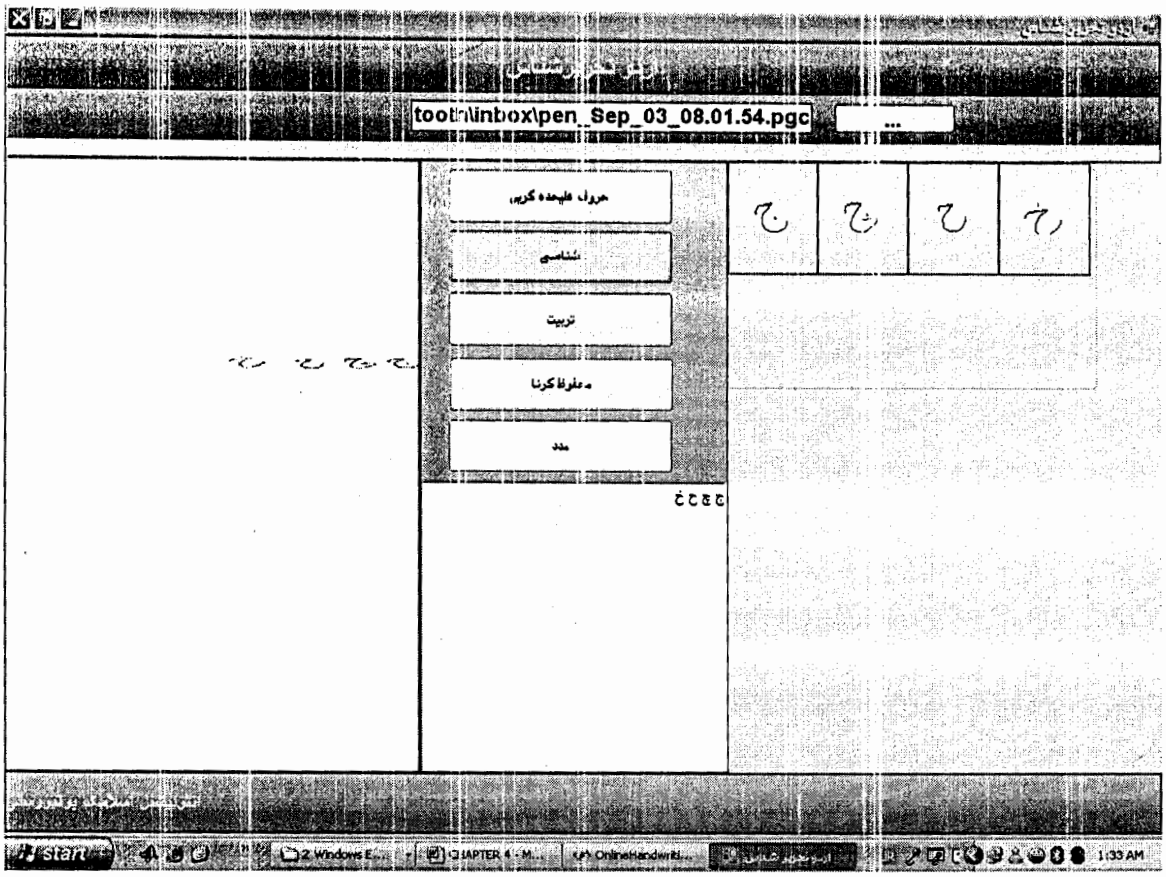

Figure 4.7: Final Result after classification

4.5.2 Recognition Process of Ligatures خ،ح،چ،ج.

4.5.2.1 Input:

The input was taken from digital pen as shown in figure 4.1 of character **مجددالف ثانی**

Figure 4.8: Input character via pen

4.5.2.2 Preprocessing:

Several preprocessing operation are performed on the input text so that some drawbacks can be removed, and correct data can be inputted for classification. The following are the preprocessing performed on the ligatures مجدالف ثانی.

4.5.2.2.1 Interpolation and Resizing

Due to the low processing and camera frame rate pen missed some points to interpolate these point interpolation is performed. As this is handwritten system, therefore written ligature have different size, to reduce variation resizing is performed.

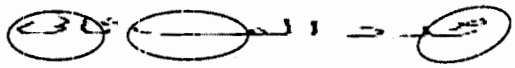

Figure 4.9: Missing data before interpolation

Figure 4.10: After interpolation and resizing of Figure 4.9

4.5.2.2.2 Segmentation of Ligature and strokes:

The primary and secondary ligatures segmented after interpolation, in the following each primary stroke with its respective secondary strokes, then segmented the primary and secondary stroke.

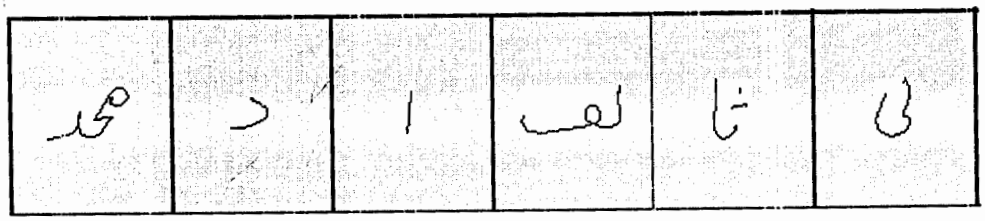

Figure 4.11: Segmentation of Primary Stroke

No	Ligature	Secondary Stroke
1	محد	Ono Secondary Stroke
2	د	No Secondary Strokes
3	ا	No Secondary Stroke
4	لف	No Secondary Stroke
5	ثا	Two Secondary Stroke
6	نی	One Secondary Stroke

Table 4.3: Segmentation of Secondary Strokes

4.5.2.3 Feature Extraction:

Preprocessing is also performed on the feature matrix to remove some wrong selected features. The following is the feature matrix selected for ligatures ج،چ،ح،خ after preprocessing on feature matrix.

No	Ligature	Feature Matrix
1	محد	11,22,33,5
2	د	8,8.33,5
3	ا	7,7,9
4	لف	7,7,9,10,35,3
5	ثا	7,7.14,3
6	نی	7,7,25,27,2

Table 4.4: Feature Matrix

4.5.2.5 Formation of Ligature by using Grammar:

Both output combined through grammar to from correct ligature. Grammar is given bellow. In this example محد have only one secondary ligature at bottom center so it forms مجد while د have no secondary stroke thus it forms د.Similarly لفhave no secondary ligature but there is no such character exist therefore the system recognized this ligature as لف , i.e one ligature on second character.

4.5.2.4 Classification.

The above feature matrix was fed to the HMM for classification. The output is shown below.

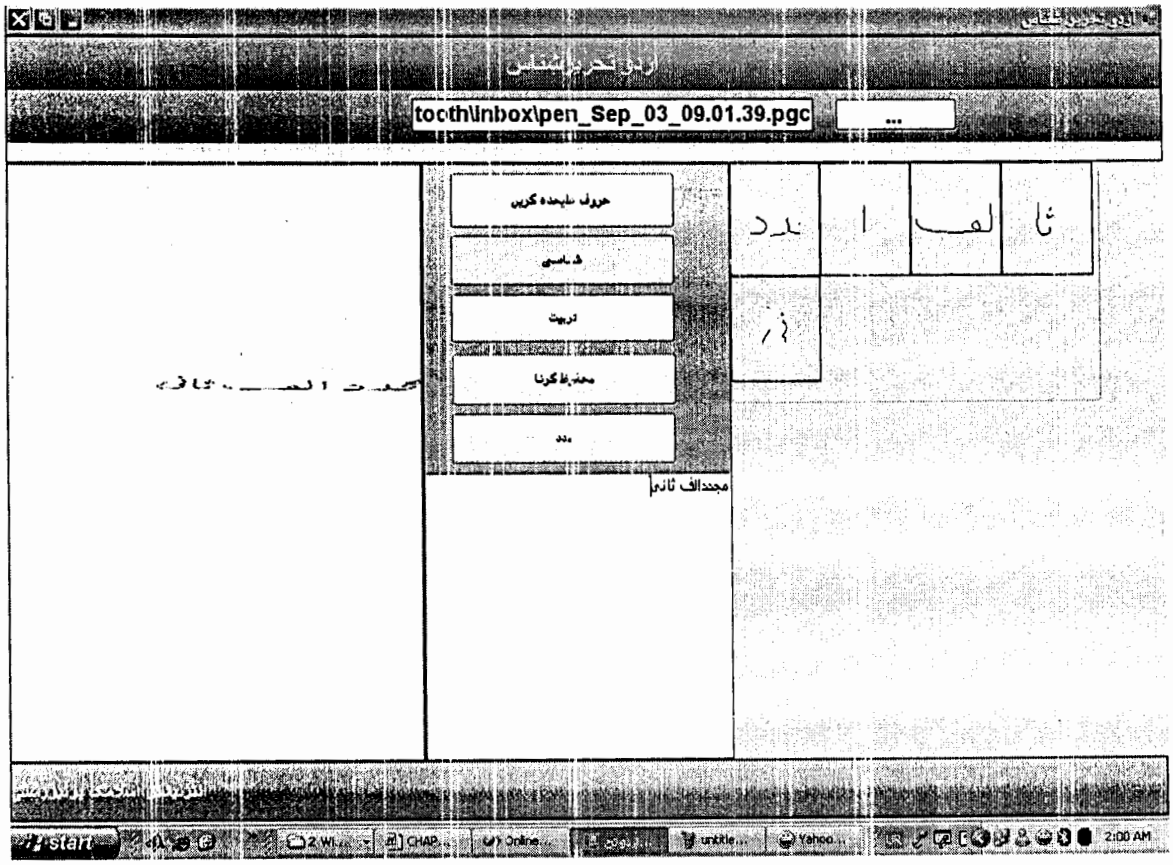

Figure 4.12: Final Result after classification.

4.5.3 Recognition of صد ،ضد and صدٔ:

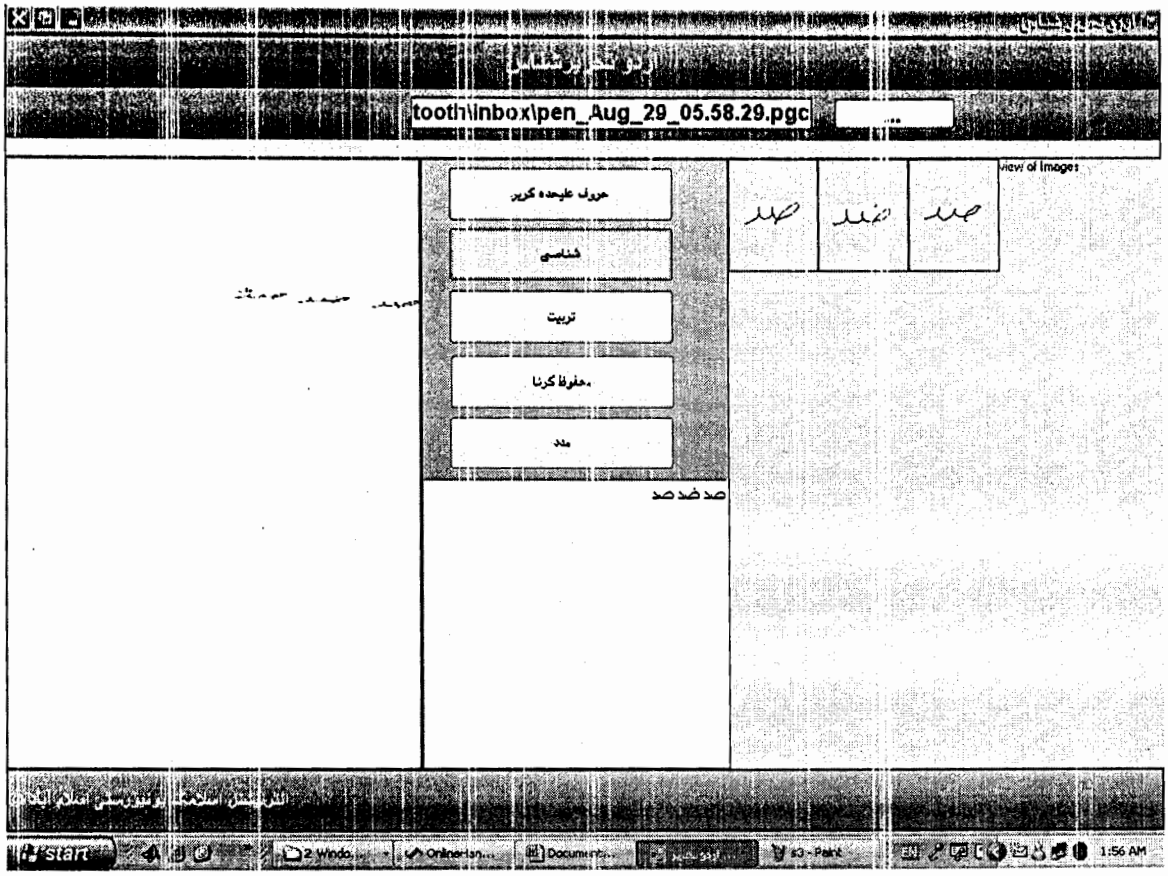

Figure 4.13: صد ،ضد and صدٔ Recognized Character

In the above testing صد ،بضد and صدٔ were tested, all primary ligature were classified correctly while secondary stroke 'ط' is not detected during the segmentation therefore the ligature صدٔ is recognized as a ligature صد.

4.5.4 Recognition of اب اوربات لب رات شام:

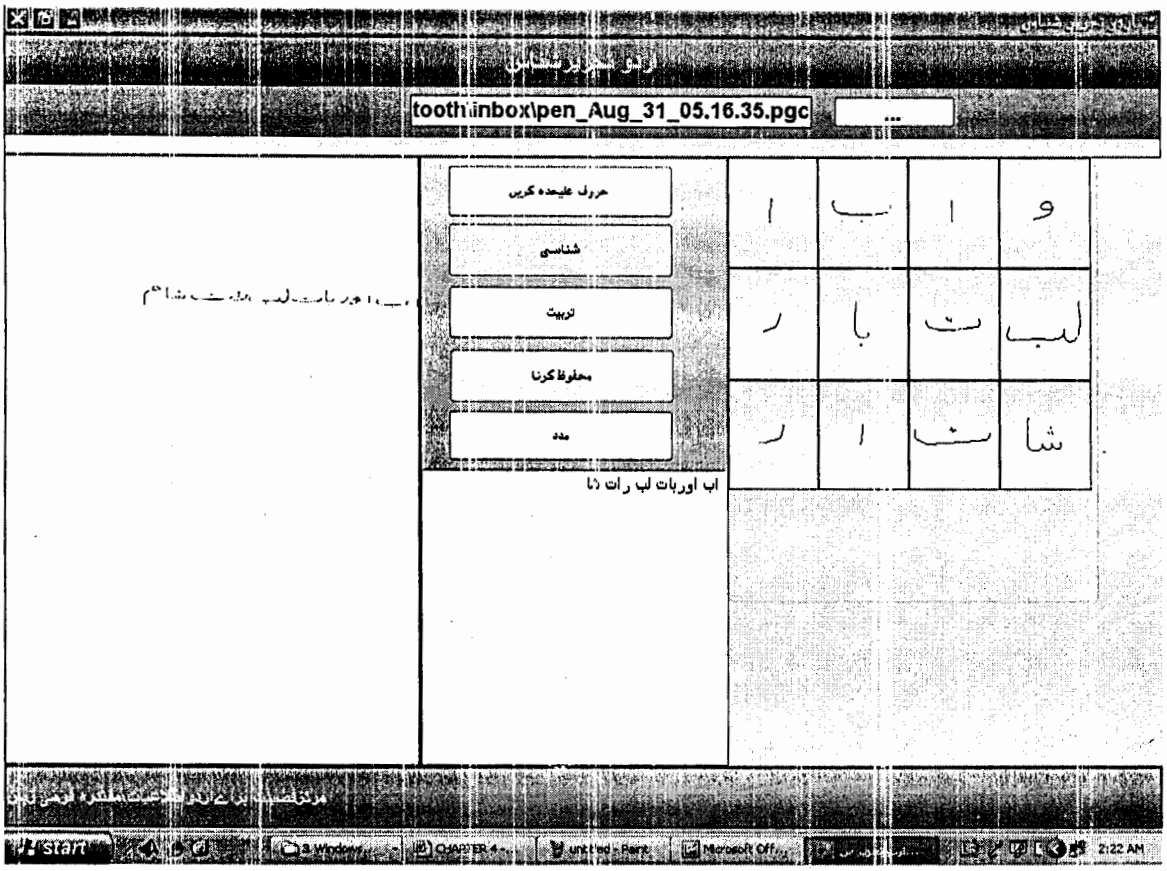

Figure 4.14: Recognition of اب اوربات لب رات شام

In the above testing ' **اب اوربات لب رات شام**' ligature are tested, all primary ligature were classified correctly expect 'م'. Actually this is the problem in segmentation of ligatures, not at the end of training.

The recognized result of above trained sample is اب اوربات لب رات شا.

4.6 Invalid Input

4.6.1 Invalid Primary Stroke:

If the primary stroke is not recognized then the system move to the next stoke, as the primary stoke recognition depends upon the last and first feature state, upon the basis of starting and ending of the ligature, class is identified, otherwise ligature will not be recognized. If the correct class is identified then system give the ligature which is very close.

4.6.2 Invalid Secondary Stroke:

If the primary stroke is correctly identified, then secondary strokes are identified, if secondary strokes are not recognized, then system moves next to the grammar phase by leaving. In the grammar phase both outputs are combined to form the correct ligature. If secondary strokes are missing, then the system insert the secondary stroke own self depending upon the ligature.

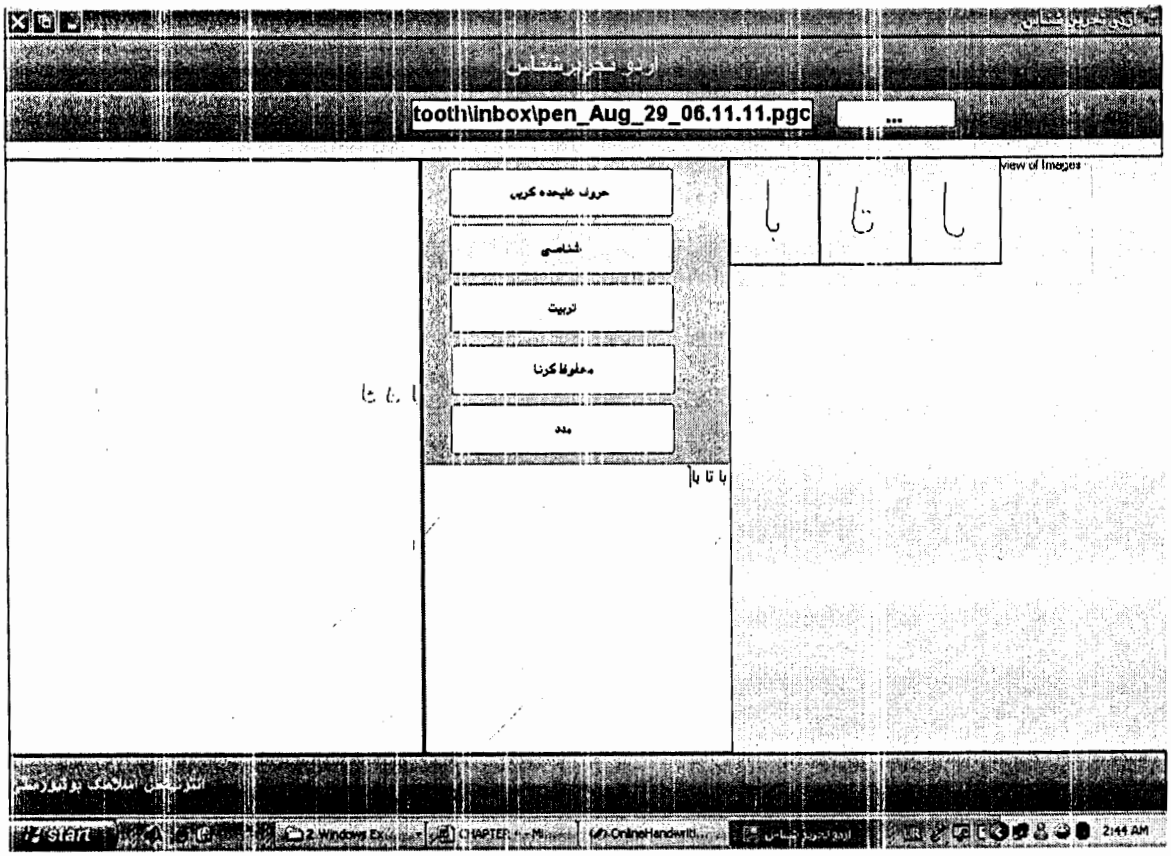

Figure 4.15: Secondary Stroke Conflict

In the above بتا بتا example user wrote while secondary stroke in the third ligature تا was not recognized, while the out put of this is بتابتا.

4.7 Constraints:

As our system is more sensitive at start and end so starting and ending of the ligature must write properly so that proper class can be identified. Speed should be normal and features like loop, jeem, and ayen should write properly. Open loops greater than the threshold are not entertained. Secondary Stroke should write carefully according to their position and size, while timing of secondary stroke does not matter.

4.8 Contribution:

In the field of online Urdu character recognition contributions are:

- The recognition of 500 two-three and four character ligatures.
- The development o the software that successfully recognizes these ligatures.
- The system is intelligent with respect to ligature formation using secondary stroke, because it does not depends upon odder of secondary stroke.

4.9 Future Directions:

As this is a research and implementation was an initial step. Therefore a large work still has to be done to increase the ligatures and recognition result.

- Work on segmentation of primary ligature and secondary ligature and preprocessing phase still require some work.
- Increase the no of ligatures, up to all ligature of Urdu so that whole language can be recognized.
- Recognition of additional secondary strokes such as the shad, zeer, zabar and paish.
- Recognition of Urdu numerals.

References

[1]. Khorsheed S.M. *"IRecognising handwritten Arabic manuscripts using a single hidden Markov model"* Pattern Recognition Letters, Oct 2003, pp 2235 - 2242

[2]. L.M. Lorigo, V. Govindaraju, "*Offline Arabic handwriting recognition: a survey*", Pattern Analysis and Machine Intelligence, Volume 28, May 2006 pp 712 - 724

[3]. Bunke H. and Wang P.S.P (1997), "*Handbook of Character Recognition and Document Image Analysis"* World Scientific Publishing Company, pp 397-420.

[4] MMM Fahmy, S Al Ali ,*"Automatic Recognition of Handwritten Arabic Characters Using Their Geometrical Features"* Studies in Informatics and Control Journal,2001.

[5]. B. Verma et.al , *"A feature extraction technique for online handwriting recognition".* IEEE International Joint Conference, July 2004, pp1337 – 1341.

[6] S.A Hussain et.al "*Design of an OCR System for Online Urdu Input" IMVV 2007.*

[7] T.Sudo "*On-Line Recognition of Handwritten Text Based on Hidden Markov Model" School of Inforamtion Science,Japan adanced institute of science and Technology .* (2002),

[8] Kosmala A., Rigoll G. "*Recognition of Online Handwritten Formulas" , IEEE Dec 2004.*

[9]. Shu H. , "*On-Line Handwriting Recognition Using Hidden Markov Models"* , Fujitsu Sci Tech J. 1996.

[10] H.TANAKA et al , "*Online Handwriting Recognition Technology and Its Applications"* Fujitsu Sci Tech J (2004), pp170-178

[11]. Mitsuru M., AKIRA N., "*Sub Stroke Approach to HMM-based On-line Kanji Handwriting Recognition" , Nakai, 200.*

[12]. Vural E., Erdogan H. "*An Online Handwriting Recognition System For Turkish" IEEE, Dec 2004.*

[13].Bunke H. and Wang P.S.P (1997), "*Handbook of Character Recognition and Document Image Analysis"* , World Scientific Publishing Company, pp 397-420.

[14]. L. Rabiner, B. Juang, *"Fundamentals of Speech Recognition",* Prentice Hall, New Jersey (1993), 506p..

[15]. *Rabiner, L. R.. "A tutorial on hidden Markov models and selected applications in speech recognition" . IEEE, VOL. 77, No. 2, February 1989.*

[16] B. Al-Badr and S. Mahmoud. "*Survey and bibliography of arabic optical text recognition.*" Signal Processing, pp 44-49, 1995.

[17]. (2006, September 8 – last update), Available: http://en.wikipedia.org/wiki/Pakistan (Accessed: 2006, September 11)

[18]. (2006, August 1 – last update), Available: http://ur.wikipedia.org/wiki/ (Accessed: 2006, September 11).

APPENDIX 1

ا	آ					
ب	پ	ت	ٹ	ث	ک	گ
ج	چ	ح	خ			
د	ڈ	ذ				
ر	ڑ	ز	ژ			
س	ش					
ص	ض					
ط	ظ					
ع	غ					
ف	ق					
ل						
م						
ن	ں					
و						
ہ						
ی	ے					

Table A1.1 List of Alphabets of Urdu

لا					
با	پا	تا	ٹا	ثا	
جا	چا	حا	خا		
صا	ضا				
سا	شا				
ما					
فا	قا	کا	گا		
عا	غا				
طا	ظا				
ہا	نا	ہا			
یا					
ھا	تھا	بھا	پھا	نھا	پھا
ببب	بپپ	بتت	بٹٹ	بٹ	
پپپ	پپپ	پتت	پٹٹ	پٹ	
تب	تپ	تتت	تٹٹ	تٹ	
ٹب	ٹپ	ٹتت	ٹٹٹ	ٹٹ	
ٹب	ٹپ	ٹتت	ٹٹٹ	ٹٹ	
جب	جپ	جتت	جٹٹ	جٹ	
چب	چپ	چتت	چٹٹ	چٹ	

Table A1.2 List of Ligatures combination of two characters

تذ	تڈ	تد
ٹذ	ٹڈ	ٹد
ڈذ	ڈڈ	ڈد
جذ	جڈ	جد
چذ	چڈ	چد
حذ	حڈ	حد
خذ	خڈ	خد
سذ	سڈ	سد
شذ	شڈ	شد
طذ	طڈ	طد
ظذ	ظڈ	ظد
عذ	عڈ	عد
غذ	غڈ	غد
فذ	فڈ	فد
قذ	قڈ	قد
لذ	لڈ	لد
کذ	کڈ	کد
گذ	گڈ	گد
مذ	مڈ	مد
نذ	نڈ	ند
ہذ	ہڈ	ہد
یذ	یڈ	ید
ئذ	ئڈ	ئد
بز	بڑ	بر
پز	پڑ	پر
تز	تڑ	تر
ٹز	ٹڑ	ٹر
ڈز	ڈڑ	ڈر
جز	جڑ	جر
چز	چڑ	چر
حز	حڑ	حر
خز	خڑ	خر
سز	سڑ	سر
شز	شڑ	شر
صذ	صڈ	صد
ضذ	ضڈ	ضد
طز	طڑ	طر
ظز	ظڑ	ظر
عز	عڑ	عر
غز	غڑ	غر
کز	کڑ	کر

گز	گڑ	گر
لز	لڑ	لر

					لا
	ٹا	ٹا	تا	پا	با
		خا	حا	چا	جا
				ضا	صا
				شا	سا
					ما
		گا	کا	قا	فا
				غا	عا
				ظا	طا
			با	نا	با
					یا
پھا	نھا	پھا	بھا	تھا	ھا
	بٹ	بٹ	بت	بپ	بب
	پٹ	پٹ	پت	پپ	پب
	تٹ	تٹ	تت	تپ	تب
	ٹٹ	ٹٹ	ٹت	ٹپ	ٹب
	ٹٹ	ٹٹ	ٹت	ٹپ	ٹب
	جٹ	جٹ	جت	جپ	جب
	چٹ	چٹ	چت	چپ	چب
			مز	مڑ	مر
			نز	نڑ	نر
			بز	بڑ	بر
			یز	یڑ	یر
			ئز	ئڑ	ئر
	ٹس	پس	تس	پس	بس
	ٹش	ٹش	تش	پش	بش
		خس	حس	پس	جس
		خش	حش	چش	جش
			شش	سش	سس
				غس	عس
				عش	عش
				قس	فس
				قش	فش
				گس	کس
				گش	کش
				لش	لس

				مش	مس
				نش	نس
				یش	یس
				ہش	ہس
				ئش	ئس
			ٹص	پص	بص
			ٹض	پض	بض
				فض	فص
				قض	قص
				لض	لص
				نض	نص
				یض	یص
				ہض	ہص
				ئض	ئص
				شط	سط
			خط	چط	جط
ٹط	ٹط	تط	پط	بط	نط
				قط	فط
				غط	عط
					ہط
					یط
					ئط
				شظ	سظ
			خظ	چظ	جظ
	ٹظ	ٹظ	تظ	پظ	بظ
				قظ	فظ
				غظ	عظ

				ہظ
				یظ
				ئظ
ٹع	ٹع	تع	پع	بع
ٹغ	ٹغ	تغ	پغ	بغ
			ضع	صع
			ضغ	صغ
			قع	فع
			قغ	فغ
			گع	کع

			گغ	کغ
			لغ	لع
			مغ	مع
			نغ	نع
			ہغ	ہع
			یغ	یع
			ئغ	ئع
ڈف	ٹف	تف	پف	بف
			شف	سف
			ضف	صف
			ظف	طف
			قف	ڤف
			گف	کف
				لف
				نف
				ہف
				یف
				ئف
ڈق	ٹق	تق	پق	بق
	خق	حق	چق	جق
			شق	سق
			ظق	طق
			قق	ڤق
			گق	کق
				لق
				مق
				نق
				ہق
				یق
				ئق
ڈل	ٹل	تل	پل	بل
	خل	حل	چل	جل
			شل	سل
			ظل	طل
			قم	ڤم
			گم	کم
				لم
				نم
				ہم

				یم
				ئم
ثن	ٹن	تن	پن	بن
	خن	حن	چن	جن
			شن	سن
			ظن	طن
			غن	عن
			قن	فن
			گن	کن
				لن
				من
				نن
				ہن
				ین
				ئن
ثو	ٹو	تو	پو	بو
	خو	حو	چو	جو
			شو	سو
			ضو	صو
			ظو	طو
			غو	عو
			قو	فو
			گو	کو
				ہو
				مو
				ہہ
				یہ
				ئہ
ثی	ٹی	تی	پی	بی
	خی	حی	چی	جی
			شی	سی
			ضی	صی
			ظی	طی
			غی	عی
			قی	فی
				لی
				می
				نی
				ہی

ی				
ئی				
بے	پے	تے	ٹے	ٹے
جے	چے	حے	خے	
سے	شے			
صے	ضے			
طے	ظے			
عے	غے			
فے	قے			
کے	گے			
لے				
مے				
نے				
ہے				
ئے				

جخ	جچ	جج
چخ	چچ	چج
حخ	حچ	حج
خخ	خچ	خج
سخ	سچ	سج
شخ	شچ	شج
صخ	صچ	صج
ضخ	ضچ	ضج
کخ	کچ	کج
گخ	گچ	گج
لخ	لچ	لج
مخ	مچ	مج
نخ	نچ	نج
بخ	بچ	بج
یخ	یچ	یج
ئخ	ئچ	ئج
	بذ	بڈ
	پذ	پڈ

Table A1.2 List of Ligatures combination of two characters

میں	مین	مٹن	متن
تنب	تیب	نیب	بیت
نبا	تنا	پنا	بنا

بیا	بیا	بیا	بیا
بٹا	بٹا	بٹا	بٹا
پٹا	پٹا	پٹا	پٹا
تنا	تنا	تنا	تنا
ٹنا	ٹنا	ٹنا	ٹنا
ٹنا	ٹنا	ٹنا	ٹنا
ٹیا	ٹیا	ٹیا	ٹیا
ٹند	ٹند	ٹند	ٹند
تبد	بتد	بتد	تبد
پند	نبد	نبد	پند
صنڈ	صبڈ	صبڈ	صنڈ

Table A1.3 List of Ligatures combination of three characters

APPENDIX 2

This project implemented in MATLAB 7 and Visual C#.Net, some of the important code is given below.

A2.1 Implementaion of Ligature 'بب'

```
# region Beb 29
        /// <beb بب   /// </summary>
        /// <param name="legature"></param>
        /// <returns></returns>
        public char[] grammar_29(Legature legature)
        {
            int d = 0;
            int oldlocation = 0;
            int dobdot = 0;
            int dund = 0;
            char[] results = new char[2];
            results = new char[] { 'ه', 'ب',' ' };

            for (int i = 1; i < legature.StrokeCount; i++)
            {
                Stroke tempStroke = legature.GetStroke(i);
                SecondaryStroke ss = new SecondaryStroke(tempStroke);
                int location = legature.getSecondaryStrokeLocation(i);
                cont = cont + 1;
                if (cont==2)
                    cont=0;
                switch (ss.Recognize())
                {
                    case SStroke.DoubleDot:
                       if (location == 6 && oldlocation == 0)
                              { results[0] = 'ى'; }
                        else if (location == 3 && oldlocation == 0)
                              { results[0] = 'ت'; }
                        else if ((location == 1 || location == 2) &&
                              oldlocation == 0) { results[1] = 'ت'; }
                        else if ((location == 6 && oldlocation == 6))
                              { results[0] = 'پ'; }
                        else if ((location == 4 || location == 5) &&
                                 (oldlocation == 4 || oldlocation==5))
                              { results[1] = 'پ'; }
                        else if (location == 3 && oldlocation == 3)
                                    { results[0] = 'ث'; }
                        else if ((location == 1 || location == 2) &
                              (oldlocation == 1 || oldlocation == 2))
                              { results[1] = 'ث'; }
                              oldlocation = location;
                              dobdot = 1;
                        break;
                     case SStroke.Tuein:
                        if (location == 1 || location == 2)
                              { results[1] = 'ٹ'; }
                        if (location == 3) { results[0] = 'ٹ'; }
                       break;
                     case SStroke.SingleDot:
                        if ((location == 6 || location == 5 || location
```

```
                        == 4) && oldlocation==0 )
                        { results[1] = 'ب'; }
                    else if ((location == 3 )&& oldlocation==0)
                        { results[0] = 'ن'; }
                    else if ((location == 6  && oldlocation == 6))
                            { results[0] = 'ی'; }
                    else if (location == 3 && oldlocation==3)
                        { results[0] = 'ت'; }
                    else if ((location == 1 || location == 2) &&
                        (oldlocation == 1 || oldlocation == 2))
                            { results[1] = 'ت'; }
                    // for double dot first and then sencond like
                    in pe
                    else if (location == 3 && oldlocation == 3 &&
                            dobdot==1)
                            { results[0] = 'ث'; }
                    else if (location == 6 && oldlocation == 6 &&
                             dobdot == 1)
                           { results[0] = 'پ'; }
                    else if ((location == 4 || location == 5) &&
                            (oldlocation == 4 || oldlocation ==
                            5 )&& dobdot == 1)
                            { results[1] = 'پ'; }
                    else if ((location == 1 || location == 2) &&
                        (oldlocation == 1 || oldlocation == 2) &&
                        dobdot == 1) { results[1] = 'ث'; }
                        oldlocation = location;
                        dobdot = 0;
                        oldlocation = location;
                    break;
                    case SStroke.Dunda:
                    if (location == 3)
                        { results[0] = 'ک'; }
                    else if (location == 2)
                         { results[1] = 'ک'; }
                        if (location == 3 && dund == 1 &&
                            oldlocation == 3)
                            { results[0] = 'گ'; dund = 1; }
                    else if (location == 2 && dund == 1 &&
                        oldlocation == 2)
                         { results[1] = 'گ'; dund = 0; }
                        dund = 1;
                        oldlocation = location;
                    break;
            }
        }
        return results;

    }
        #endregion
```

A2.2 Chain Code Implementation

```
public int[] calculateChainCode(PGC_Parser.PenStroke ps)
        {

            Point[] pnts = ps.Points;
            int[] chaincode = new int[pnts.Length - 1];
            string str = "";
            for (int i = 0; i < pnts.Length - 1; i++)
            {
                int diffx = pnts[i].X - pnts[i + 1].X;
                int diffy = pnts[i].Y - pnts[i + 1].Y;
                if (diffx == 1 && diffy == 1)
                    chaincode[i] = 4;
                else if (diffx == 0 && diffy == 1)
                    chaincode[i] = 3;
                else if (diffx == -1 && diffy == 1)
                    chaincode[i] = 2;
                else if (diffx == 1 && diffy == 0)
                    chaincode[i] = 5;
                else if (diffx == -1 && diffy == 0)
                    chaincode[i] = 1;
                else if (diffx == 1 && diffy == -1)
                    chaincode[i] = 6;
                else if (diffx == 0 && diffy == -1)
                    chaincode[i] = 7;
                else if (diffx == -1 && diffy == -1)
                    chaincode[i] = 8;
                str += chaincode[i];

            }

            return chaincode;
        }
```

A2.3 Decoding Implementation

```
function[Ligname]=decoding(OChainCode,LigT,LigE,Tr,Lig)

NPath=LigE;

  if Tr==1
            [Ox ,Oy]=size(OChainCode);
            [ui uo]=size(LigT);
            FinalPat=zeros(Ox,30);
            EstimateO = multibandread(LigE,[50,40,1], 'double',0,'bsq','ieee-le');%Read
            Probablities from HD
            EstimateT = multibandread(LigT,[50,50,1], 'double',0,'bsq','ieee-le');
             for multi=1:1:Ox
                 for jj=1:1:30
                    if OChainCode(multi,jj)>0
                       OCha(jj)=OChainCode(multi,jj);
                    end
                 end
           [FinalPath VetProb MaxProb] = viter_HMM(OCha,EstimateT ,EstimateO); %
                 [sx sy]=size(FinalPath);
                 FinalPat(1,1:1:sy)=FinalPath(1,1:1:sy);
                 FinalPaths(multi,1:1:30)=FinalPat(1,1:1:30);
                 FinalPath=0;
                 FinalPat(1,1:1:sy)=0;
            end

            for j=1:1:uo
                 if NPath(j)=='.';
                    NPath(j+2)='_';
                    break;
                 end
            end
  end
end
```

User Manual

User Manual

HMM Based Online Handwritting Recongition works with Anoto Digital Pen. The Anoto technology Pens are available with most popular brand names of Nokia, Logitech and Maxell. The software is compatible with the following Digital Pen hardwares

- Nokia Digital Pen SU-1B
- Nokia Digital Pen SU-27W
- Logitech io2 Digital Pen
- Maxell Digital Pen
- Hitachi Digital Pen

The pen should support pgc file format version 1.1. Anoto Pen works with especially designed paper using Form Development ToolKit.

Digital Pen has a camera and Infrared device. When the writer preses the pen the camera and IR device starts capturing the location on the paper. The 0.3mm dotted box on the paper enables the pen to understand the location on which the Pens button is pressed.

Positioning Of Pen On Paper

The printed paper should be place on a smooth and hard surface. The Pen should be placed straight on the Paper and can start writing.

Writing Style

The following are the guideness for writng the urdu text.

- First write primary stroke and then secondary strokes relevant to that primary stroke.
- Start and end of the primary ligature must be written carefully.
- The direction must be from right to left except some ligature like ج.
- Dot should be written on write position exact over that character.
- Writing speed must be normal.
- Writing text must be horizantly on the page.

- Secondary stroke should be wrtten with respect to the occurance of the character in the base ligature like in بتا , first seconday stroke signle dot of bey and then secondary stroke double dot of tey should be written.

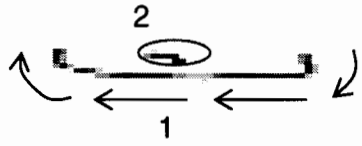

Figure A3.1 Writing direction of the strokes

Example

The figure A3.1 shows the writing direction and the timing of the strokes. The direction of ligature bey is from write to left and secondary stokes are give after the primary stroke. The secondary stroke must be the pen down and pen up, not to press the pen for a time at the same position, because pen conineously get the points and this will lead to invalid secondary stroke.

After writing the text send the written text to the computer via bluetooth after writing the text. Open the software and clike on the button on the top right corner to open the pgc file, locate that pgc file this will show the pgc file on the left side. Click on the button 'حروف علیحدہ کریں' to see the seperation of ligatures, and click on the 'تربیت' button to recognized the input pgc file. The system writes the recognized file in the text area, it can be copied or save this file press the button with tag ' محفوظ کرنا ' as a document file.
Invalid character or missing character can be corrected by clicking the text area and then write the ligature which is not recognized.

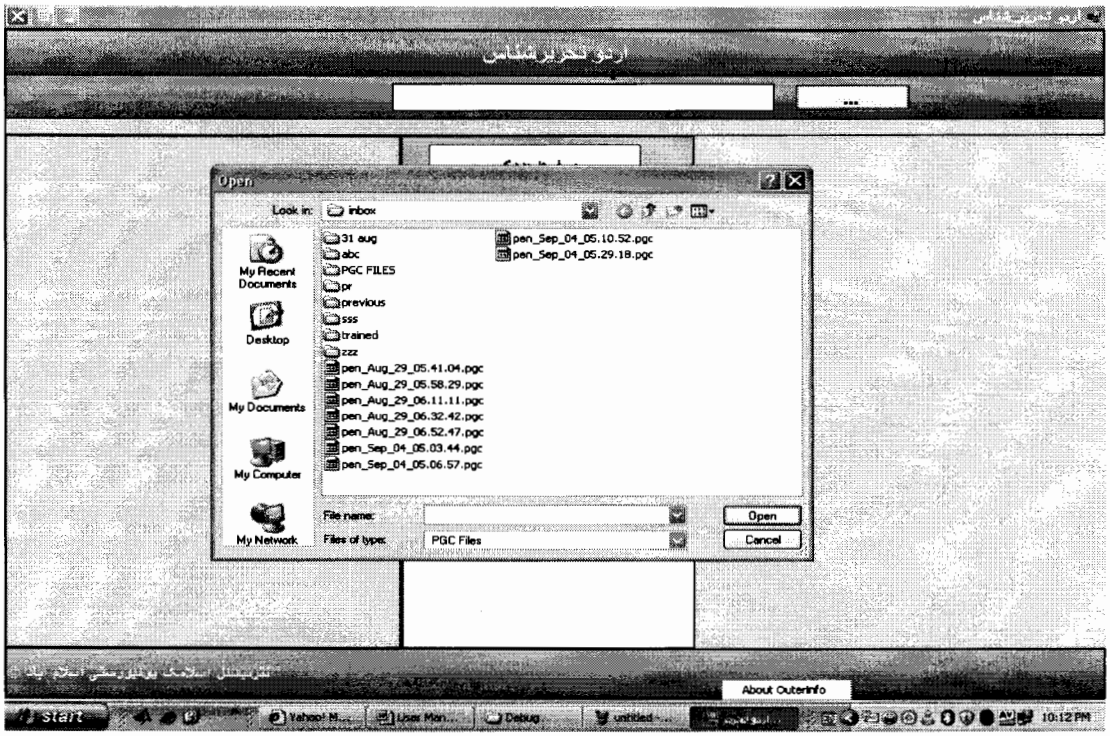

Figure A3.2 Reading the PGC File

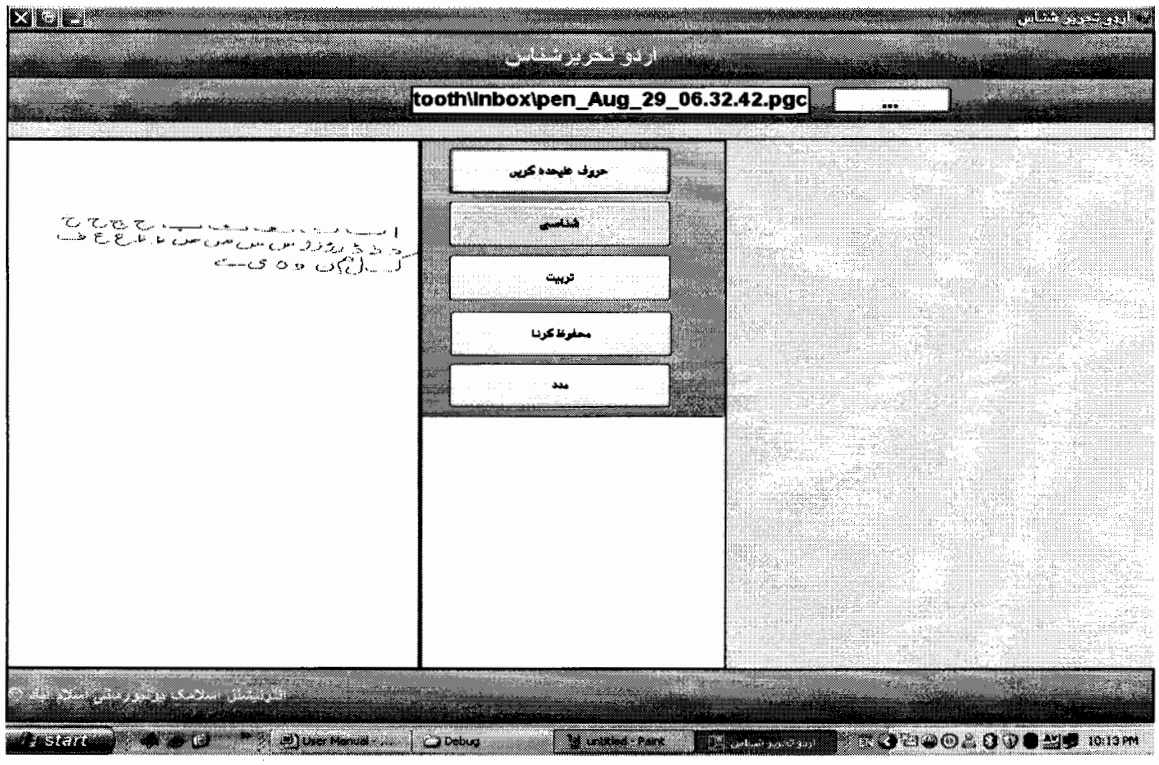

Figure A3.3 Seperation of Primary and Secondary Stroke

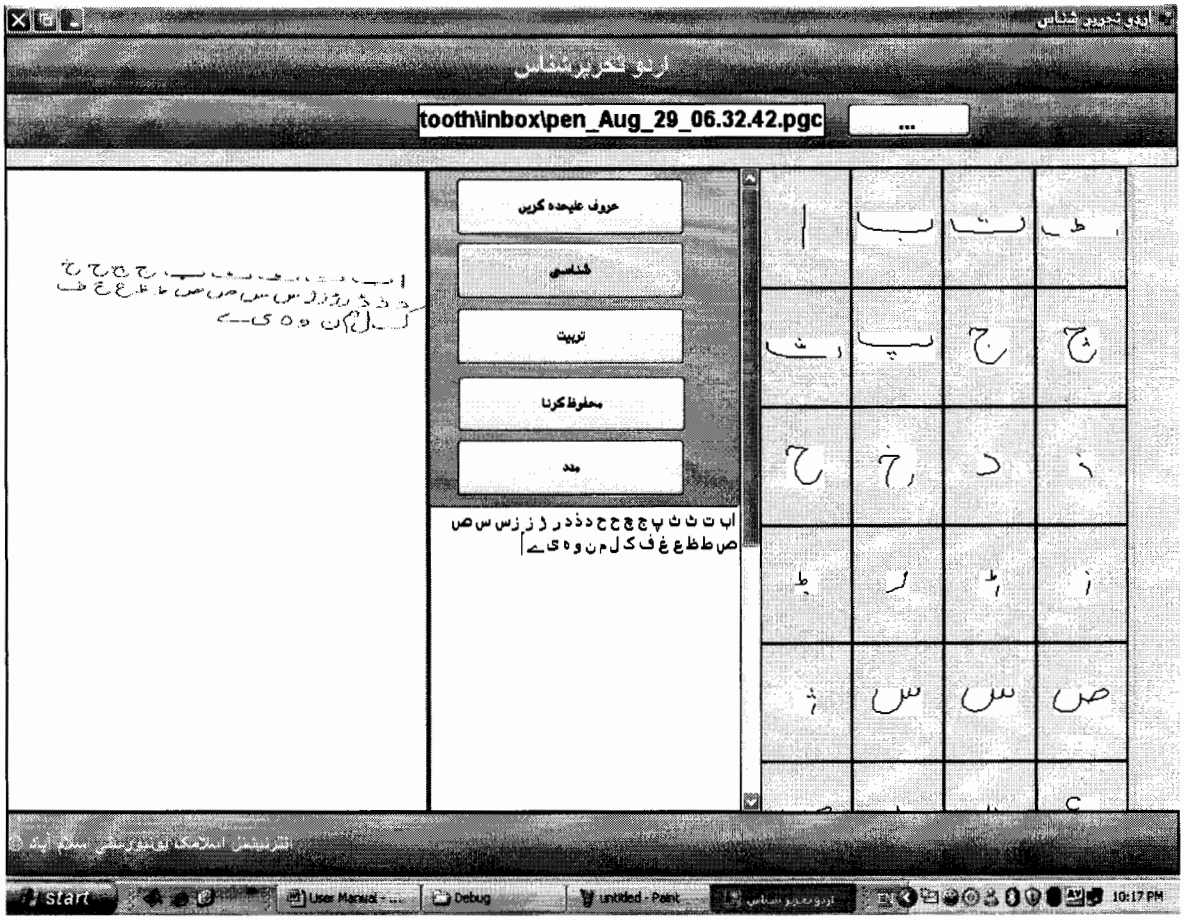

Figure A3.4 Final recognized text

RESEARCH PAPER

HMM Based Online Urdu Character Recognition

Faculty of Basic & Applied Science
International Islamic University,
Islamabad.

Abstract: *Urdu online handwriting recognition is a very difficult task because it is naturally cursive. This paper introduces a Hidden Markov Model based technique that provides solution for the complicated Urdu script online handwritten recognition. This system makes use of ligature based approach. The analysis of the Urdu is future difficult due to the segmentation of primary and secondary stroke that are compulsory for Urdu script. We used the feature based approach and selected twenty six unique feature.We only used the Nasta'liq font because Urdu is typically written in Nasta'liq style. This is the first HMM based solution to the Urdu online handwritten recognition.*

Keywords: Urdu, HMM, Ligature based OCR, Online Handwriting ICR, Urdu Handwriting Recognition

1. Introduction.

Online handwritten recognition has been an on-going research problem for near on four decades. It has been gaining more interest due to the increasing popularity of hand-held computers, digital notebooks and advanced cellular phones. Conventionally, communications between man and machine have been based on keyboard and electronic mice or pointing device. These methods can be very inconvenient when the machine is only slightly bigger or same size. More and more efforts are being made on the software as well as on the hardware side in order to make this human computer interaction friendly. The pen interfaces is a key element in providing an efficient and natural way of input. Two such natural alternatives to typing are handwriting and speech recognition which are natural communication methods.

There are two strategies that are implemented in cursive script recognition are Holistic strategies in which the whole word is recognized i.e. ligature base recognition and Analytic strategies in which segmentation is performed. Holistic strategies (Method based on probabilistic distance) implemented in this proposed system. Automatic handwritten recognition has been classified into two categories based on the input data: online and offline. Offline handwritten recognition does not require immediate interaction with the user while online handwritten recognition has completely interaction with the user. The root of online handwriting recognition is real time data collection by way of a digital sampling method. The most common input devices are digitizing tablets or digital pen, where the written data is digitized and translated into a series of coordinates.

In this paper we introduce online handwriting recognition for the Urdu script. More than 160 million people can speak Urdu and also it is the national language of Pakistan [2].

2. Charestrics of Urdu Script.

Urdu script consisted 36,37, 40, 53 or 58 basic letters. Urdu script is written in cursive style from right to left in both handwritten and machine printed. Urdu is also the context sensitive language and written in the form of ligature. Most letters have different shapes depending on their position in the word e.g. the letter appeared as isolated, middle, center, and end. Urdu script uses the punctuation markers to separated words. Characters overlap each other while writing the Urdu text. Urdu words have primary and secondary stroke. The primary stroke (ghost character) is main ligature and secondary strokes are the diacritical marks i.e. dots, zeer, zeber, hey act.

As handwritten is more complex than printed text, because more variation in written text. Thus recognition for handwritten Urdu is much more complex than any other language like English. The complexities of Urdu language as compared to other languages are Context sensitive shape, Cursiveness, Overlapping , Secondary stroke for each main stroke, Baseline, Ligature , and space between the ligature.

3. Previous Work.

There are many offline OCR systems available for handling printed Arabic and Persian, Urdu documents with reasonable levels of accuracy. However, very negligible work done on Urdu online handwritten recognition. This may be due to the complexities involved in the online character recognition with the added difficulties of Urdu script handwriting. There are basically two techniques for recognizing words. One is the segmentation based which involves the division of a word in to its sub parts i.e in to individual characters. Other is the segmentation free or ligature based recognition, in which the word is recognized as a whole without trying to segment it in to characters.

S.Malik and S.A.Khan, [8] have recognized only individual characters (Urdu alphabets) and Urdu numerals (0-9) in their research but ligatures have not been addressed. Using the individual characters, only 200, two character words were recognized. For example, اب, در, etc. The recognition rate for the isolated characters and numerals is 93% and 78% for two character words.

Al–Emami and Usher [m] devolved an online Arabic handwriting recognition system based on decision tree technique. This system was tested with 13 Arabic- letter shapes [5].

S. Asma, A.Fareeha S. A. Husain presented a method for recognition of online Cursive Urdu hand written *Nasta,liq* Script. The system was trained for 250 ligatures. By using multiple classes of features, they have improved the number of ligatures that can be identified. That system can successfully recognize 250 base ligatures and 6 secondary strokes. These when combined form more than 864 ligatures which can recognize approximately 50000 words of our Urdu dictionary successfully[6].

The above methods are not sufficient for the task of recognizing of Urdu script. The system presented by Asma Sajjad, Fareeha Anwar S. A. Husain can recognized only 864 ligature from which three character ligature are only 53 and mostly recognized ligature are not useable in the Urdu. Also they did not work on the diacritical marks like zeber , zeer, ect. Their system could not recognize the ligature up to four character ligature.

4. Proposed System.

OLUR (On-Line Urdu Recognizer) Proposed system use of the various approaches in order to recognize the strokes. This is due to the cursive nature of the Urdu handwriting. We have used the segmentation free i.e. ligature based approach in which the input stroke is not broken in to characters as many of the recognition errors occur due to errors in segmentation and complexities of segmentation especially for handwritten text. The segmentation free system extracts features for each ligature which is then passed on to the Hidden Markov Model for classification of the ligature. Using the strokes (x, y) co-ordinates and the chain codes, we extracted twenty six unique features for primary strokes.

We build separate HMMs for ghost character which have 50 states and 40 observation symbols and separate HMMs for secondary stroke which have 5 states and 10 observation symbols. Ghost characters are first recognized and then we recognized the secondary stroke relevant to that primary stroke and then combine both outputs to form the valid ligature. Grammar is developed for each ligature because when ghost ligature combined with secondary stroke forms different ligature.

The special ligatures are identified from the base ligatures. These special ligatures are associated with the base ligature.

5. Preprocessing.

As there exist vast variety of writing styles among user in writing the Urdu text. Moreover some users have different speed of writing. To overcome these drawbacks preprocessing is applied on the input. In this section we describe our approach in term of geometric processing, smoothing, de-hooking, segmentation of primary and secondary stroke.

Geometrical processing is performed to normalize the written text like scaling and rotation so that handwritten variation is minimized.

Figure 3: Before Preprocessing problems (a) صا (b) طا.

Due to the low processing power and low camera frame rate pen skips some points it depends upon the writing speed, to compute these points interpolation was performed. More over for better result the writer should write with normal speed, otherwise its a chance that pen will miss some important features like loop, cusp.

Figure 4: Missing data before interpolation

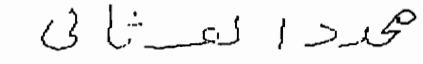

Figure 5: After interpolation and resizing on Figure 4.

Hence obtained data often contains irregularity such as the hooks and erratic handwriting generated by inexperienced users. Hooks are very common artifacts found at the beginning and ends of the strokes. Hooks are generated during fast writing, when inaccuracies during pen up and pen down while placing the pen on, or lifting it off. These often create problems in the detection of the original ligature. De-hooking also created some problem because small up at the starting of the character jeem 'ج' was removed if there was no hook at the starting of this ligature.

As the written text have zigzag path due to user hand shivering by nature. Smoothing is one of simplest approaches for data filtering. We performed two to three pixels smoothing on the chain codes of the stroke as per the variations in the chain codes of the.

Segmentation of Stoke.

The basic rule is that any Urdu character has one main stroke (ghost character) and zero or more secondary strokes or delayed strokes as shown in figure 1. Segmentation of primary and secondary strokes is essential to distinguishing among various Urdu letters. Segmentation of primary and secondary stroke is very difficult task.

Figure. 4. a. zero secondary stroke, b. One secondary stroke.

In consequence the handling the primary and secondary stroke is crucial for proper recognition of the Urdu script. In Urdu delayed stroked are written below or above the main ligature and could appear before, after or within the ligature. We developed algorithms to segment this delayed stroke. This algorithm first recognized the stroke and then finds their position respectively.

The detection of secondary stroke performed is based on the location and timing information. We divided the ligature into six blokes i.e. ligature length is divided into three parts beginning, enter and end and also height is divided into three parts below , up and mid and then find the position of the secondary stroke.

Feature Extraction:

We extracted twenty unique features; these were extracted by using the time information which occurs first as shown in figure 5. Feature matrix is build and preprocess is also applied on this feature matrix so that unnecessary feature was removed for better classification.

Start-Eight Chain Code:

This feature depends upon the starting of the ligature either diagonal, left, up act like اب.

Start Vertical Down:

This feature was selected when the ligature was a straight vertical downward in the beginning. For e.g.ط،ل.

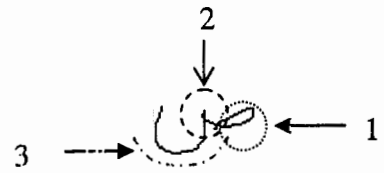

Figure 5: Feature extraction with time

Start Vertical Up:

This feature was selected when the ligature was a straight vertical upward in the beginning. As there was no word which starts from upward but this feature is used to differentiate numerals like ١ and ١ are written with same style. So for numerals we write from downward.

End Vertical Up:

This feature was selected when the ligature was a straight vertical upward in the end. For e.g. با، کا طا;

End Vertical down:

This feature was selected when the ligature was a straight vertical downward in the end. For e.g. بم م.

Horizontal RtoL:

This feature was selected if the during writing the ligature the pen movement is from right to left horizontally For e.g. in ب، ف.

Diagonal RtoL:

This feature was selected if the during writing the ligature the pen movement is from right to left diagonally like in ک.

Diagonal LtoR:

This feature was selected if the during writing the ligature the pen movement is from left to right diagonally like in لا.

Horizontal LtoR:

If the during writing the ligature the pen movement is from Left to right horizontally then the Horizontal lefttoright is selected. For e.g. in ے، سے .

Hedge RtoL:

In Urdu characters like Noon, Seen, Qaf have we can say semi circle sort of shape in them. For such characters we have selected a feature called the hedge. This feature is selected when semi circle present from right to left like ق، ن.

Hedge LtoR:

In Urdu characters like Jeem, Ayen have we can say semi circle sort of shape
in them. For such characters we have selected a feature called the hedge. This feature is selected when semi circle present from right to left like ج ع .

Curve LtoR:

The direction of writing of these curves varies from right to left and also from right to left. Therefore, Curve LtoR has been selected for characters those writing direction is right to left like ن ر .

Curve RtoL:

If the curve direction of the character from left to right then Curve RtoL is selected like ع ج سے .

Cusp:

A cusp is a sharp turning point in a stroke. This feature is selected for the ligature which contains the cusps such as those present in Seen and Seen Ray like س سر .

Intersection:

When ever an intersection is encountered in a stroke this feature is selected for that stroke. For e.g. these are present in Seen, Tuwn etc. like ط فل.

Ray/Dal:

This feature is selected for the character ray of Urdu alphabet. If any ligature is a combination of ray or dal then this feature is also selected for that particular ligature like ر،بد،د .

Loop Up:

In order to differentiate the loop in fay, Qaf this feature was identified and selected. The writing direction of the loop in Qaf is clockwise so we selected Loop up feature like ق ف.

Loop Down:

In order to differentiate the loop in fay, Qaf and Meem, this feature was identified and selected for Meem. The writing direction of the loop in Meem is anti clock wise shown in figure below. Therefore, this loop down was selected for meem like بم،جم تم.

Loop Swad:

In order to differentiate the loop in fay, Qaf , Meem and Swad, this feature was identified and selected for Swad. As the swad loop is like an egg shape so it is identified to separate the Swad from other loop like ص بص.

Hey:

In order to differentiate the loop in fay, Qaf and Meem and hey , this feature was identified and selected for Hey like ہ -

BayRay:

This feature was selected if character some character combine with ray like جر بر .

Bayyee:

This feature was selected if character some character combine with choti ye like جی بی

Aien Bit:

This feature was selected if character Aein is detected in any ligature like ع .

Hay middle Bit

This feature is selected in the presence of hay in the ligature like جہل سہل .

Tuan Bit:
This feature is selected on the presence of tuan in any ligature like ط، سط بطا.

Madaa:
All the Madaa ligatures have this feature selected for them. The shape of madaa and other madaa ligatures for example آ ، آ .

6. Classification:

A HMM is doubly stochastic model and the underlying stochastic process corresponds to state transition that are hidden but the state changes are observed through another set of stochastic process that produces the out put symbols[12]. In this experiment we used the discrete HMMs were used for the classification of Handwritten Urdu and left to right topology was used.

The training data is done by taking input from Urdu literate trainer and testing this system for online Urdu handwritten script on the selected words. For training purpose input data is acquired from 15 people so that more variations are trained.

No	Ligature	Recognition Rate
1	~	93 %
2	.	100 %
3	/	96 %
4	..	98 %
5	ط	95 %
6	ء	95 %
7	؏	94 %

Chain codes are used to represent a boundary by a connected sequence of line segments. In this paper the chain codes are taken for the features of characters. We use the 8-connectivity is used to minimize the loss of points.[7].

Hidden Markov Models (HMMs) are finite state machines and a powerful statistical models for modeling sequential or time-series data, and have been successfully used in many tasks such as speech recognition, OCR, information extraction and robotics.

There is no known approach to solve the model parameter that maximizes probability of observation. The Baum-Welch training algorithms is used for parameter estimation $\lambda = (A, B, \pi)$ for each ligature that best describe the data.

7. Conclusion and Future work:

In this paper, we introduce a HMM based method for recognition of online Cursive handwritten Urdu Nastaliq Script. The system is currently trained for 800 ligatures. This system minimizes the errors due to segmentation free approach. By using multiple classes based features, we have improved recognition result. For testing purpose input was taken from 10 trained users and every ligature is tested. The system provided accuracy approximately 75%.

As this research and implementation was an opening step in handwriting Urdu character recognition. Therefore, there is a lot of scope for future enhancement.

- Improvement in preprocessing technique so that correct feature will be extracted.
- Increase the no of ligature so that whole Urdu can be recognized.
- Recognition of additional secondary strokes such as the shad, zeer, zabar and paish.
- Dictionary for word classification.

No	Ligature	Total Samples	Recognition Rate
1.	ا class	10	100
2.	ب class	10	95
3.	ج class	10	83
4.	س class	10	87.7
5.	ص class	10	88.3
6.	یا	10	95.8
7.	طا	10	88.5
8.	صا	10	84.3

7. Reference:

[1]. (2005,November 10 – last update), Available: http://www.nlauit.gov.pk (Accessed: 2007, Feb. 3).

[2]. (2006, September 8 – last update), Available: http://en.wikipedia.org/wiki/Pakistan (Accessed: 2006, September 11).

[3]. (2006, September 8 – last update), Available: http://en.wikipedia.org/wiki/Pakistan (Accessed: 2006, September 11).

[4]. Mohammad S. Khorshed, William F. Clocksin, "Structural features of cursive Arabic script", proc of 10th British Vision, Conference, University of Nottingham, UK, September-1999.

[5].Samir Al-Emmy and Mike Usher, "On-Line Recognition of Handwritten Arabic Characters", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12. No. 7. July 1990

[6]. Anwar F., Asma. Thesis , "*Online Urdu Character Recognition Engine.*" IMVE 2007.

[7]. A. Onat, F. Yildiz, and M. Gündüz, " Ottoman Script Recognition Using Hidden", Transactions on Engineering, Computing & Technology. VOL14 AUG. 2006.

[8]. Malik, S.; Khan, S.A., "Urdu online handwriting recognition", Emerging Technologies, 2005. Proceedings of the IEEE Symposium on Volume, Issue, 17-18 Sept. 2005 Page(s): 27 – 31, Digital Object Identifier 10.1109/ICET.2005.1558849.

[9]. Zahra A Shah and Farah Saleem. "Ligature Based Optical Character Recognition of Urdu, Nastaleeq Font", INMIC 2002.

[10]. A.Amin, "Machine Recognition of Handwritten Arabic Word" by the IRAC II system, 6th Int.Conf on Pattern Recognition, Munich, 1982,34-36.

[12]. Speech Technology Magazine, issue July 1999, As with Speech, "Online Handwriting Recognition Enables PCs to Understand Natural Human Input" By Eran Aharonson.