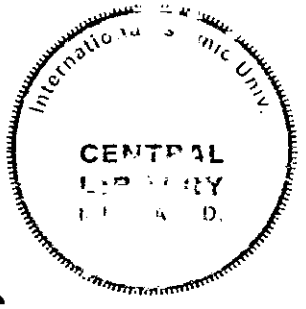
Online Urdu Character Recognition Engine (OLUCR)

Developed by

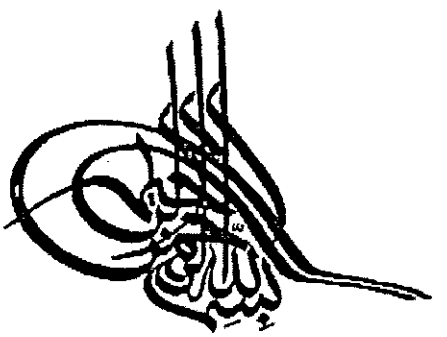
Asma Sajjad MS216
Fareeha Anwar MS233

Supervised by

Dr. Syed Afaq Hussain



Department of Computer Science
Faculty of Applied Sciences
International Islamic University, Islamabad.
(2006)



In The Name of

ALLAH ALMIGHTY

The Most Merciful, The Most Beneficent

"Recite: In the Name of your Lord Who created, created man from clots of blood. Recite: And your Lord is The Most Generous, He Who taught by the pen, taught man what he did not know. No indeed! Truly man is unbridled seeing himself as self-sufficient. Truly it is to your Lord that you will return." (Surat al-'Alaq: 1-8)

A dissertation submitted to the Department of Computer Science, Faculty of Applied Sciences, International Islamic University, Islamabad, Pakistan, as a partial fulfilment of the requirements for the award of the degree of

MS in Computer Science

DEDICATION

We dedicate this project to *our beloved country Pakistan and our National language Urdu*. The success and development of IT in Pakistan lies in the hand of the IT Professionals and IT students. This project is a step in this respect.

Asma Sajjad 216-FAS/MS/CS/F04
Fareeha Anwar 233-FAS/MS/CS/F04

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 further declared that we have developed this project and accompanied 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

Asma Sajjad 216-FAS/MS/CS/F04
Fareeha Anwar 233-FAS/MS/CS/F04

ACKNOWLEDGEMENT

All praise is to Almighty Allah, The Lord of Creation, the known and unknown universe, Who descended His last Prophet Hazrat Mohammad (SAW), the Benefactor of Humanity; to guide the mankind and enlighten the ones who believe Almighty Allah, Who bestowed us good health, courage and knowledge to carry out and complete our work.

It is unimaginable that an academic effort of this magnitude could successfully come to fruition without the help of others. Expressing gratitude to those whom it is due is a highly regarded Islamic custom based upon the statement of Prophet Muhammad (S.A.W):

"Whoever does not thank people does not thank Allah."

We express our Highest Gratitude to our Kind Supervisors Dr. Syed Afaq Hussain who kept our morale high by his suggestions and appreciation. His motivation lead us to this success, without his sincere and cooperative nature and precious guidance, we could never have been able to complete this task. We would also like to pay special thanks and our Highest Gratitude to Sir Khalid Jamil Azizi, GM Technical, Mechatronics Department, National Development Complex for providing us the tablet for such a long duration of our thesis. This project would never have been possible without his help.

We would also like to acknowledge the support of our Family members. We would like to admit that we owe all our achievements to our truly, sincere and most loving parents, brothers and sisters, who mean the most to us, and whose prayers are a source of determination for us.

Asma Sajjad 216-MS/CS/2005
Fareeha Anwar 233-MS/CS/2005

PROJECT IN BRIEF

Project Title: Online Urdu Character Recognition Engine

Under Taken By:

Asma Sajjad
Reg. No. 216-CS/MS/2005

Fareeha Anwar
Reg. No. 233-CS/MS/2005

Supervised By:

Dr.Syed Afaq Hussain
Head of Department of Telecommunication
International Islamic University, Islamabad.

Starting Date: 26 October, 2005.

End Date: 30 June, 2006.

Tools and Techniques:

- Visual C# dot net.
- Graphic Tablet: Intuos Wacom board.
- MS Windows 2000

ABSTRACT

The reduction in the prices of technology has increased the use of handheld devices. These devices also provide the facility of text input but the small keys on them do not provide a convenient way of input. Digitizing tablets on the other hand provide a natural and convenient way of input. There are many online character recognizers for many languages but there is no such commercial product for Urdu. Our research deals with the online Urdu handwriting recognition.

Our online Urdu handwriting recognizer makes use of the ligature based approach instead of character based identification. The segmentation free system extracts a feature vector for each ligature which is then passed on to the back propagation neural network for classification of the ligature. The special ligatures (Dots, Tay, Hamza, Diagonal & Mad) are identified from the base ligatures. These special ligatures are associated with the base ligature. After this, the ligature is checked for its validity. Valid ligatures form words. After word formation, word validity is checked by using a word dictionary. Finally, the valid words are written in a text file.

The OLUCR recognizes all two character ligatures e.g. شا, جب etc and most commonly used three character ligatures e.g. تھا, ہیں, کیو, میں, لیا etc. We have successfully recognized 241 base ligatures and 6 secondary strokes. These when combined form 864 ligatures which can recognize approx 50000 words of our Urdu dictionary successfully. Special feature recognition is 98% and base ligature recognition is 93%.

TABLE OF CONTENTS

Chapter 1
INTRODUCTION

1.1 Introduction to OCR:

Any type of data processing by computer has many advantages in the viewpoints of cost, convenience and efficiency. Recently, people are trying to process data by computer in many fields. To realize computer data processing, it is essential to input all data into the computer. But there is a bottleneck on the input process because inefficient input methods are a burden in terms of much time, cost, and labor. As a solution of this problem, automatic optical character recognition comes to the rescue.

Optical Character Recognition is the branch of pattern recognition. It is one of the most difficult and intriguing pattern recognition problems that studies methods of converting text in images into computer understandable text codes. One of the aims of OCR is to emulate the human ability to read at a much faster rate by associating symbolic identities with images of characters. Some of the potential applications of Optical character recognition are: reading postal address off envelopes, automatically sorting mail, reading customer filled forms, archiving and retrieving text, Provision of interface for entering text in PDAs through pen like devices, etc. The ultimate goal of character recognition is to develop a communication interface between the computer and its users. This implies the direct storing of handwritten or typewritten text into the computer without using the keyboard.

We can group the research in character recognition into two main schemes, namely, Offline character recognition and Online character recognition.

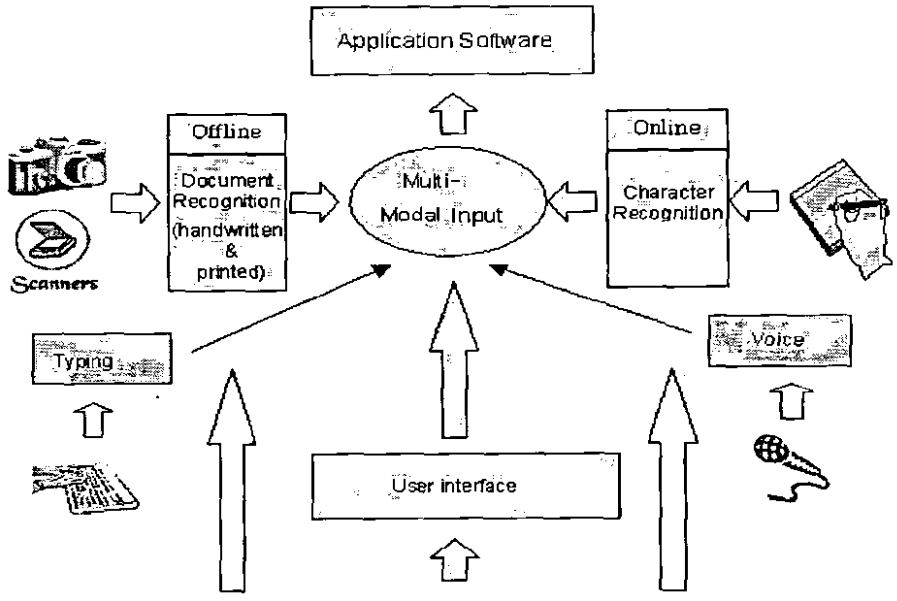


Figure 1: Online and offline OCR

1.2 Offline Character Recognition:

In offline systems, the data is collected using a scanner/digital camera whose output is usually in the form of a bitmap, image or compressed form of it. Further, the recognition is not done at the time of writing the documents i.e. in real time.

Text Recognition Systems generally have following stages: image acquisition, pre-processing, segmentation, feature extraction, classification and recognition [6]. From the classifier perspective, the Text Recognition Systems are further divided into Segmentation based and Segmentation-free systems [7].

1.2.1 Segmentation Free Systems/ Ligature Based Recognition:

In these systems, the word is recognized as a whole without trying to segment and recognize characters or primitives [15].This approach does not need character segmentation algorithms, but efficient ligature recognition algorithms. One approach for such systems is to calculate a single feature vector for each word; this feature vector is then used to recognize the word.

1.2.2 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 [5].

1.3 Online Character Recognition:

Enabling computers to understand natural human input has been the goal of many researchers in the last 2 decades. Extensive research has been done in both voice and handwriting recognition technologies. The introduction of the personal computer and PDAs has expanded the consumer market, which is now ready to acknowledge handwriting-based solutions, both on the hardware technology side and the ability of adequate online handwriting recognition. It is a very important technique for convenient human computer interface. Pen-based input gives lots of advantages. It provides the most easiest and natural way of input i.e. pen based. 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 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 touch pads, where the written data is digitized and translated into a series of coordinates.

1.3.1 Online Input Methods:

Since natural handwriting recognition is a difficult task, different methods have been tried to achieve accurate recognition. Each approach is more suitable for a certain task. The following is a list of the most popular:

Unistroke Characters:

This input consists of characters created out of a single ink stroke (Unistroke). The shape of each character is selected in a way to increase the differentiation from the other characters. The user must learn a new handwriting style (some letters are similar to standard writing) and also abide by regulations describing how to use the system, such as how to shift between capital and small letters. This method is highly suitable for extremely small devices on which it is impossible to write naturally. [14]

Boxed Input:

The input is done by implementing a natural handwriting style, but each character has to be input by using a predefined box layout. The written strokes are segmented into characters according to their location in the box. This method is usually suitable for form-filling applications. [14]

Natural Input:

This writing style is completely natural, as if the user is writing on a piece of paper. The ink strokes can be connected, such as in cursive writing, and there are no constraints in the writing style. This method is highly suitable for massive input tasks, such as E-mail, note taking, etc. [14]

Command and Control:

Handwriting input can also be used as a command and control method (similar to the usage of speech). A specific handwritten symbol can be attached to an action, such as the execution of a macro, launch of an application, performing editing operations and more. This method is highly suitable as an add-on to devices that already incorporate a touchpad (such as laptops) or within any pen-centric device. [14]

The most sought after input method is a totally natural input, in which the user writes as if writing on paper. Recently, some adept natural handwriting recognition systems were introduced, which can be used without constraints. Most natural online handwriting recognition engines share a common basic design. However the actual implementation varies quite a bit. [14]

1.3.2 Methods for Recognition:

The process of recognition of text written in any language can be broadly classified into five main categories, which are:

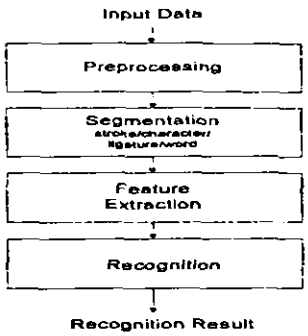


Figure 2: Recognition Process

Data Collection:

The data is collected in real time from the acquisition hardware. The standard data is a stream of {x, y} coordinates, sampled at almost equal time intervals. In addition to the coordinates, there is also an indication if the stylus is at up or down mode. Data collection ends after a predefined timeout period, where no additional pens down events are encountered.

Pre-processing:

Pre-processing of the input raw image is crucial for the success of efficient character recognition systems. This process is a collection of different operations applied on an image. It takes a raw image and enhances it by reducing noise and distortion, and hence simplifies the processing stage. Different steps involved in the Pre-Processing phase are: Skew detection and removal, Filtering, Smoothing, thinning, normalization, Document decomposition, Slant normalization etc.

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.

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.

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.

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 modelling 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.

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.

1.4 Urdu Character Recognition

1.4.1 Background:

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. About 104 million population uses Urdu as its first and second language across the globe. Urdu is the national language of Pakistan and is also widely spoken in Afghanistan, Bahrain, Bangladesh, Botswana, Fiji, Germany, Guyana, India, Malawi, Mauritius, Nepal, Norway, Oman, Qatar, Saudi Arabia, South Africa, Thailand, UAE, United Kingdom and Zambia [2].

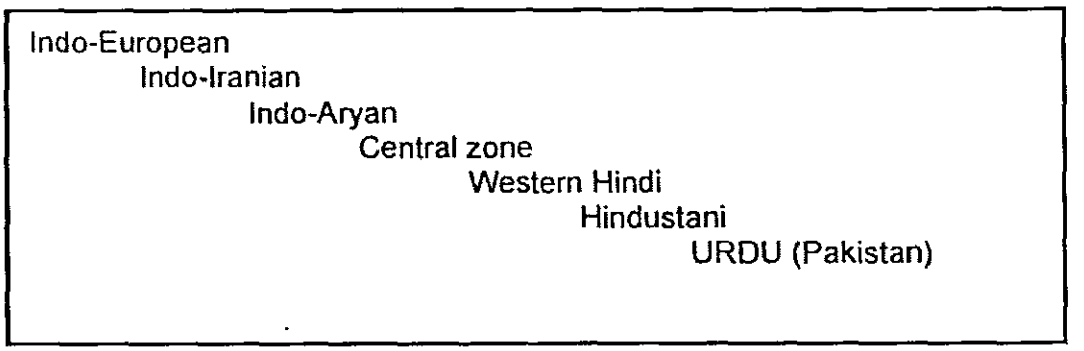


Figure 3: Language Family Tree for Urdu [1]

Formal vocabulary of Urdu is borrowed from Arabic and Persian while it embodies morphological and phonological similarities with Hindi. Perso-Arabic script written in Nastaleeq style is widely used for Urdu orthography [2].

1.4.2 Characteristics of Urdu script:

Some of the most important characteristics of Urdu, which distinguish it from other languages, are:

- **Direction of writing:** Unlike English, Urdu is written from right to left.
- **Cursive:** Urdu text is cursive in nature [10]. Adding to the complexity is the writing style in which the characters forming words are connected to each other.
- **Ligatures:** Several characters of Urdu are combined vertically to form a ligature [10].
- **Spaces:** Spaces in Urdu may occur between ligatures and between words. The spaces between ligatures and words vary. This feature of Urdu handwriting also makes the recognition difficult.
- **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**: Diacritics are very important in Urdu language. These include diacritics such as *Dots, Ttaay, Hamzaa, Diagonal and Madaa*, etc.[16]
- **Upper and Lower Cases:** There is no concept of upper case and lower case in Urdu language writing.
- **Shape of the Characters:** 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 [10].
- **Strokes:** The basic rule is that any Urdu character has one main stroke and zero or one secondary stroke [10].

1.4.3 Challenges in Urdu Script:

The problem of Urdu text recognition is closely related to Arabic text recognition. Urdu character set is based on the Arabic character set. It is a cursive language even in its printed form. In the past, a lot of research has been done on automatic recognition of text written in languages based on Roman, Chinese text, Arabic and Persian, but no serious research has ever been published on Urdu text recognition. Arabic and Persian are based on similar basic characters and writing styles as Urdu, have seen quite worthwhile research in the past decade. However, those solutions are not valid to Urdu due to a number of inherent differences in the script and styles of Urdu text. Like Arabic,

recognizing Urdu script presents challenges of cursive orthography and context sensitive letter shape.

1.4.4 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.
- Filling forms in Urdu and their digitization.
- Teaching Aids
- Touch screen data input
- Ticketing Machine
- Urdu signature verification.

1.5 Research Objective:

This research opens doors to a new dimension to the Urdu Computing. Although there has been a lot of work in Offline Urdu Recognition but Online Urdu Recognition has been unfortunate in this respect.

The objective of this research is to build a recogniser for Urdu handwriting. The final software has the capability of recognising the Urdu ligatures and words. It also discards the invalid ligatures and words. The transferring of the handwritten text to a text file is another objective of this research which is another very useful facility to the end user.

1.6 Research Contribution:

The contribution of this thesis is the first ever Online Urdu Handwriting Recogniser that can recognise 50,000 words of Urdu dictionary with a reasonable amount of accuracy.

1.7 Structure of Dissertation:

The thesis is divided in to five chapters. In addition to these five chapters three appendix have been added for additional information. The chapters and appendix cover the following areas. The chapter 1 is Introduction. It gives an introduction to the optical character recognition, it's types. It also describes the characteristics of the Urdu language and applications of online Urdu recognisers.

Chapter 2: Literature Survey

This chapter gives an overview of the work done in the field of Urdu and Arabic character recognition, as Arabic is closely related to Urdu.

Chapter 3: Proposed Solution

This chapter gives a detail of the research approach employed to the problem of online Urdu handwriting recognition.

Chapter 4: Experimental Results

In this chapter the experimental results are discussed as well as displayed by giving the screen shots of some of the written and recognised words.

Chapter 5: Conclusion

This chapter discusses the results, the constraints and the future enhancement in this research.

Appendix 1: Ligature List

A list of the 864 tested and trained ligatures is provided in this appendix.

Appendix 2: Word Dictionary

A list of some of the words present in the dictionary is provided in this appendix.

Appendix 3: Implementation

Some sample code of important modules is provided in this chapter.

Chapter 2
LITERATURE REVIEW

2.1 Introduction:

The rapid spread of computers and palm devices, has led to enormous growth in the availability of information, as well as dynamically increased demand for access to information in electronic format for purposes such as document analysis, processing and efficient editing. Giant steps have been made in the last decade, both in terms of technological supports and in software products. Character recognition (OCR) contributes to this progress by providing techniques to convert large volumes of data automatically. In order to answer the extremely tedious and time consuming task of re-inputting information into electronic media via keyboard, another field has emerged with in OCR i.e. the dynamic character recognition. There are so many papers and patents advertising recognition for the successful static character recognition i.e. offline character recognition claiming recognition rates as high as 99.99.There has been a lot of work in dynamic character recognition in English, Chinese, Japanese and Thai languages however, Arabic and Urdu languages have been unfortunate in this respect.

During our thesis, we studied a number of research papers, tools and techniques for Arabic and Urdu dynamic as well as static character recognition which helped us to identify our dimensions of research. In this chapter we have included some of the most relevant papers. Our study shows that dynamic character recognition is an emerging field that requires a lot of research in order to develop robust methods for providing commercial products with extremely high recognition rates.

2.2 Arabic Character Recognition By Adnan Amin

Published In: School of Computer Science and Engineering, University of New South Wales, Sydney, New South Wales 2052, Australia.

The objective of this paper is to identify the problem related to hand written Arabic character, and describe different methods for the recognition of hand written Arabic characters for online recognition.

Amin proposed a system for online Arabic word recognition. The hand drawing is directly segmented in to characters on the basis of certain heuristic criteria. For every word, the characters are connected to each other by horizontal segments from right to left. Statistically, theses connections appear:

1. Almost always after an intersection point.
2. Often after a cusp point.
3. Sometimes simply after a change of curvature.

These points serve remarkably as separators in the segmentation process and directly permit one to obtain a list of characters of the word component. With the help of each separator, portions of the hand writing are extracted and transmitted to the character recognition module for identification. If no character has been recognized, the corresponding segmentation is canceled and a new tentative try is carried out with the separator of the next lower priority.

The character recognition module is similar to the system which recognizes isolated characters. Finally, three hypotheses at most, i.e. the three best score candidates provided by the character recognition module are associated with each of the characters which are extracted from the handwritten Arabic word by segmentation module. The set of hypothesis form a lattice. Identification of the word consists of traversing the lattice to find the path of the best score corresponding to a word in the dictionary. Binary diagrams are used for resolving ambiguities and eliminating all a candidates which are not present in the dictionary of known words.

The second method is global without segmentation in to characters. This method uses the notion of stroke instead of character. A vector defining the main parameters of word (number of secondary strokes, size and position of groups of dots, number of intersection points, number of cusp points) makes it possible to recognize a word and each of its constituent strokes. Whenever the same vector yields several possibilities, they are classified according to a score computed from secondary parameters of the stroke (the start point, form and angular variation of the main stroke).

Furthermore to enhance the recognition rate of a syntactical and semantically analyzer that verifies the grammatical structure and the meaning of Arabic sentence is used.

Conclusion:

This paper defines both the segmentation free and segmentation based methods for online Arabic character recognition. As we choose the first approach for recognition, this paper helped to figure out the kind of features we could extract from the handwritten Urdu text. This paper was basically an extract from a book, so there were no recognition rates mentioned which could help us to identify the success of these methods.

2.3 Automatic Recognition of Handwritten Arabic Characters Using Their Geometrical Features By Maged Mohamed Mahmoud Fahmy

Published In: Computer Science Department, College of Science University of Bahrain Isa Town

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 does preprocessing, finds the feature vector and then performs classification of each character. Details are given below.

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.

Conclusion:
The writer has chosen the segmentation free approach for offline character recognition. This leads to the recognition through feature extraction. Further feed forward back propagation neural network was used for classification. They achieved 53% accuracy in first trial. In the second trial, the recognition accuracy increases to 69.7%. This paper gave us an insight to the power of neural networks for character classification for large data.

2.4 A Multi-tier Holistic approach for Urdu Nastaliq Recognition By Syed. Afaq Husain and Syed. Hassan Amin

Published In: IEEE INMIC Dec. 2002, Karachi.

In this paper, a new approach for the off-line recognition of cursive Urdu text has been presented. This methodology has been developed for the Noori Nastaliq Script. Word (Ligature) based identification has been adopted instead of character based identification. A multi-tier holistic approach is used to recognize ligatures from a pre-defined ligature set. Initially, the special ligatures (Dots, Tay, Hamza & Mad) are identified from the base ligatures. These special ligatures are associated to the most probable neighboring base ligature in the second step. Finally, the above information along with some other RTS invariant features of base ligature is presented to the Feed Forward Back Propagation neural network to perform the final recognition task. The different stages involve:

Preprocessing:
The preprocessing stage involves Smoothing, Skew detection and correction, Document decomposition, Slant normalization etc.

Segmentation:
They have applied Connected Component Labeling. This technique assigns to each connected component of binary image a distinct label. The labels are usually natural numbers from 1 to the number of connected components. The algorithm scans the image from left-to-right and top-to-bottom. On the first line containing black pixels, a unique label is assigned to each contiguous run of black pixels. For each black pixel, the pixels in its eight neighborhoods are examined, if any of these pixels has been labeled the same label is assigned to the current pixel, otherwise a new label is assigned to it.

Feature Extraction I:
In this stage, the features that help in the recognition of special ligatures are extracted. These features are Solidity, Number of Holes, Axis Ratio, Eccentricity, Moments, Normalized segment length, curvature, ratio of bounding box width and height.

Special Ligature Identification:
For identifying special ligatures, a Feed Forward Back propagation neural network with 15 inputs, 25 hidden and 25 output neurons was used. The feature vectors obtained from Feature extraction 1 stage of the system are fed to this neural network. It then identifies the ligatures as either special ligatures or base ligatures.

Feature Extraction II:
In this stage, association of special ligatures with the base ligatures takes place.

Future enhancements in this paper can be made by increasing the number of trained ligatures and diacritics.

Conclusion:

The performance of this system was 100% on trained ligatures. The untrained ligatures were given the closest match. They have again used the bp neural network for recognition which further enhances its importance in the field of character recognition. In this and all other papers different types of preprocessing steps were performed before character recognition. We deduced from this fact that pre-processing is an essential preliminary step in all types of character recognition.

2.5 Ligature Based Optical Character Recognition of Urdu, Nastaleeq Font By: Zahra A Shah and Farah Saleem

Published In: Department of Computer Science, Kinnaird College for Women, Lahore, Pakistan, Multi Topic Conference, 2002. INMIC 2002, Dec. 27-28, 2002.

This work addresses the problem of recognizing Nastaleeq script of Urdu Language. The input to the system is an image of a page of text and the output is a machine editable file compatible with Unicode standards. The system first extracts lines from the document image and then isolates every ligature in that line. The final recognition is achieved by template matching. The system works reasonably well for type written Nastaleeq script. The basic methodology of this OCR system is:

Image Acquisition & Storage:

After the image is acquired, its contents are stored as ones and zeros in a 2 dimensional array.

Separate Lines:

Simple methods for separating lines are those based on projection profiles and the same have been used using additional information of line width in order to include secondary strokes too.

Isolate Ligatures:

The first step towards ligature isolation is to label every connected component in the line. Then mathematical features of every connected component in a line are calculated which include height, width, Aspect Ratio and Bounding Box of every component. The Aspect Ratio of the main body also forms a part of the Feature Matrix.

Distinguish between Main Bodies and Visible Features:

Visible features include: diacritics, secondary strokes and Nuktas (dots). In this step horizontal projection to construct a hypothetical band across a line of text is used. Further, a two-pass method for recognition of visible features is used. In the first pass, the visible feature set is recognized and in the second pass the visible feature recognition strategy is applied to the main body set, so as to extract the visible features coming in the region of the base band. The visible features are recognized using template matching.

Get Constituent Unicode characters:

After identifying the class to which a ligature belongs to, its constituent Unicode characters are retrieved from the Database and appended in a text file. If a class has more than one ligature, then the main body of the unknown ligature is matched with the templates of this class and the ligature with maximum similarity is selected. Its constituent Unicode characters are looked up and appended in a file.

In this paper formatting like spaces and tabs are not handled in the converted text. Further a font size of 36 performs well when aspect ratio is used as a feature vector.

Changing the font size has cause problems in identifying some ligatures when used with aspect ratio. This paper has potential for future enhancement.

Conclusion:

The techniques mentioned in this paper such as the projection profile and connected component labeling is meant for offline character recognition. This paper was helpful with respect to its Unicode retrieval and appending Urdu text to a file.

2.6 A Feature Extraction Technique for Online Handwriting Recognition By Brijesh Verma1, Jenny Lu1, Moumita Ghosh2, Ranadhir Ghosh2

1 Faculty of Informatics & Communication, Central Queensland University, Rockhampton, QLD 4702, Australia

2 Schools of Information Technology and Mathematical Sciences, University of Ballarat, Ballarat, VIC 3350, Australia

Published In: IJCNN 2004: International Joint Conference on Neural Networks (Budapest, 25-29 July 2004).

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, feature extraction, stroke extracting, calculate PEN-UP, extract 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 method and techniques proposed in this paper have shown improvement when compared with previously existing techniques.

Conclusion:

This is the first paper on online character recognition. In this paper we learnt the pre-processing techniques for the real time character recognition. They have presented a new approach to feature extraction i.e. the feature extraction through zoning. The recognition rates are 98.2% for digits, 91.2% for uppercase and 91.4% for lowercase.

2.7 On-Line Recognition of Handwritten Arabic Characters By Samir Al-Emmy and Mike Usher

Published In: IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12. No. 7. July 1990

This research paper deals with online handwritten Arabic words recognition. It recognizes a very small set of words which includes only four words of Arabic language which are used as the post code in many Arab countries. These words involve (حي، محله، دار، زقاق).

The recognition Modules involves the following stages.

Pre-Processing:
This stage involves the segmentation of the word written. Each segment is assigned a code w.r.t. it's direction. Four directions North, South, East and West i.e. 1, 3, 4 and 2 have been used. Along with this the segment length and the tangent value representing the slope of each segment is also calculated. Special code the 0 code is inserted for dots.

Learning:
This stage involves the storing of the information calculated in the above stage in the decision tree data in the form of codes. Therefore, each decision tree consists of the code number of the word, the dots information both for the upper and lower dots, and the number of character having the same dot.

Recognition:
This stage involves the tree traversal for the code of the newly entered word starting from the initial code. If the code is found as it, a very successful match is found. Otherwise, the tree traversal starts again by dropping the initial character of the code. If the dots flag is zero, the record will not be checked for the presence of dots. If it is set then dots will also be identified.

This research still has a lot of potential for further improvement as the key to online handwritten character recognition is to deal with the variations with in the same words.

Conclusion:
There are very few papers available for real time character recognition. This paper introduces an approach suitable for only few mentioned characters. This approach can be taken forward because it will definitely confuse the additional characters with the existing. Further, a new approach to pre-processing has been introduced which might be helpful in future. The results of this research were 100 % accurate only on the proper, well- written words. In case of little variation the results were incorrect.

2.8 Urdu Online Handwriting Recognition By Shumaila Malik and Shoab A. Khan

Published In: 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.

In this paper only individual characters and Urdu numerals are recognized but ligatures have not been addressed. Using the individual characters, 200, two character words were recognized. For example, اب, در, etc. In many on-line character recognition papers, for other languages, neural networks were used for recognition but they have used tree based dictionary search for the classification of characters.

The working of recognition system is based on

- analytical approach to segmentation for feature extraction

- rule based slant analysis for slant removal
- tree based dictionary search for classification

Conclusion:

This is the first ever published work on online Urdu character recognition. This paper uses the segmentation approach in order to recognize characters. The classification of characters is done using the tree based matching. This paper deals with a very small set of words and only characters not ligatures. The recognition rate for the isolated characters and numerals is 93% and 78% for two character words. This paper uses a different approach for character recognition which was not very helpful in our research. In fact, this paper was found after the completion of our research, and has been included only for comparison and as a reference to the previous work in this field.

2.9 Literature Survey Conclusion:

In order to find out the present trends and goals of online Urdu character recognition a literature survey was carried out. During the literature survey of our thesis we found out that there are many online character recognition systems and research papers for English language for example **Off-line Character Recognition using On-line Character Writing Information** by Hiromitsu Nishimura and Takehiko Timikawa Dept. of Information and Computer Sciences, Kanagawa Institute of Technology. There is also quite a number for Chinese and Thai language for example **Thai Online Handwritten Character Recognition Using Windowing Back propagation Neural Networks** by Sutat Sae-Tang Ithipan Methaste Information Research and Development Division, National Electronics and Computer Technology Center, National Science and Technology Development Agency.

There are a number of papers on off-line Arabic character recognition that have been mentioned in our literature survey. There are very few systems for Arabic online character recognition that have the capability of recognizing the whole of Arabic character and ligature set. A lot of work has been done for off-line Urdu character recognition. Some of the papers mentioned in our literature survey are **A Multi-tier Holistic approach for Urdu Nastaliq Recognition** by Syed. Afaq Husain and Syed. Hassan Amin Published in IEEE INMIC Dec. 2002, Karachi and **Ligature Based Optical Character Recognition of Urdu, Nastaleeq Font** by Zahra A Shah and Farah Saleem Department of Computer Science, Kinnaird College for Women, Lahore, Pakistan Multi Topic Conference, 2002. INMIC 2002. When it comes to research on online Urdu character recognition Urdu language has been quite unfortunate in this respect, as only two research papers were found on this subject. There are a number of papers on offline Urdu character recognition, some of which have been mentioned in our literature survey.

2.10 Problem Definition:

Online character recognizers for many languages are present. For e.g. English, Japanese, Chinese, Thai, Arabic but only two research papers have been published for Urdu yet. Urdu is a complex language when it comes to recognition by computer. It is

difficult to recognize because of having quite distinct and complex characteristics such as:

- It has a variety of scripts and styles.
- It is a cursive script.
- It has a vast character set.
- Its characters have a context sensitive letter shape.
- It has overlapping characters.
- It has a vast special ligature set which has to be associated with the base ligature.
- In a single base ligature there are often more than one secondary strokes, which increases the complexity.

Urdu like Arabic is a cursive language. Arabic character set does match with that of Urdu but Urdu tends to have a larger character set. Arabic character set consists of 28 characters where as Urdu consists a total of 42 characters [16]. In spite of the similarities between the character sets of the two languages they have a different dictionary, different ligature sets to recognize. Therefore, techniques used for the recognition of Arabic text can not be totally applied to Urdu character recognition.

Hence, the online Urdu character recognition remained unsolved. In order to achieve the online Urdu character recognition a new research has been carried out keeping in view the challenges presented by the cursive Urdu script and its context sensitive letter shape.

Chapter 3
PROPOSED SYSTEM
Online Urdu Character Recognition Engine

3.1 Introduction:

The recognition of handwritten or printed text by computer is referred to as OCR i.e. optical character recognition. When the input device is a digitizer tablet that transmits the signal in real time as in pen-based computers and personal digital assistants or includes timing information together with pen position it is called the dynamic recognition. When the input device is a still camera or a scanner, which captures the position of digital ink on the page but not the order in which it was laid down, it is called the static or image-based OCR. Dynamic OCR is an increasingly important modality in Human Computer Interaction.

Data entry using a pen forms a natural, convenient interface especially for handheld devices, which are very common now. The large number of writing styles and the variability between them makes the problem of writer-independent handwriting recognition a challenging pattern recognition problem. The structural approaches has long been dominating the online character recognition (OLCR) technology, in which the structure of input character is extracted and either matched with the structure of models already stored in a model database or to form a feature vector to determine the class of input character. The input is the pen or stylus moving across a digitizer yielding a stream of points. These are typically x, y coordinate pairs. A single pen down to pen-up movement is called a stroke. It is these strokes on which the whole of the recognition process depends. The recognition can be done through neural network or any other approach of pattern recognition. The success of the final product is its recognition rate. A higher recognition rate and the ability to support various writing styles is what make a standard recognition engine successful.

3.2 Online Urdu Character Recognition Engine:

The recognition engine makes use of the various approaches in order to recognize the strokes. This is due to the cursive nature of the Urdu handwriting. The recognition systems are generally divided in to two types. Segmentation based and segmentation free recognition systems. We have used the segmentation free approach in which the input stroke is not broken in to characters as many of the recognition errors occur due to errors in segmentation. 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 segmentation free system extracts a feature vector for each by 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.

The OLUCR recognizes 38 one character ligatures , 726 two character ligatures e.g. شا, جب etc and approximately 100 most commonly used three character ligatures e.g. تھا, ہیں, کیو, میں, لیا etc. These when combined can recognize approximately 50000 words of our Urdu dictionary successfully. The secondary strokes recognized are:

Figure 3.1: Secondary Strokes

Namely, (left-right order) hay, kaaf and gaaf long diagonal stroke, Madaa, Hamzaa, and the single dot. The special ligatures (Dots, Tay, Hamza, Diagonal & Mad) are identified from the base ligatures. These special ligatures are associated with the base ligature. After this, the ligature is checked for its validity. Valid ligatures form words. After word formation, word validity is checked by using a word dictionary. Finally, the valid words are written in a text file for display and further editing as per user requirement.

3.3 Tools and Technologies:

This system was has been developed using the following Tools and Technologies were used:

- Graphic Tablet: Intuos Wacom board.
- Visual C # dot net
- MS Windows 2000
- Urdu Dictionary.

3.4 Block Diagram

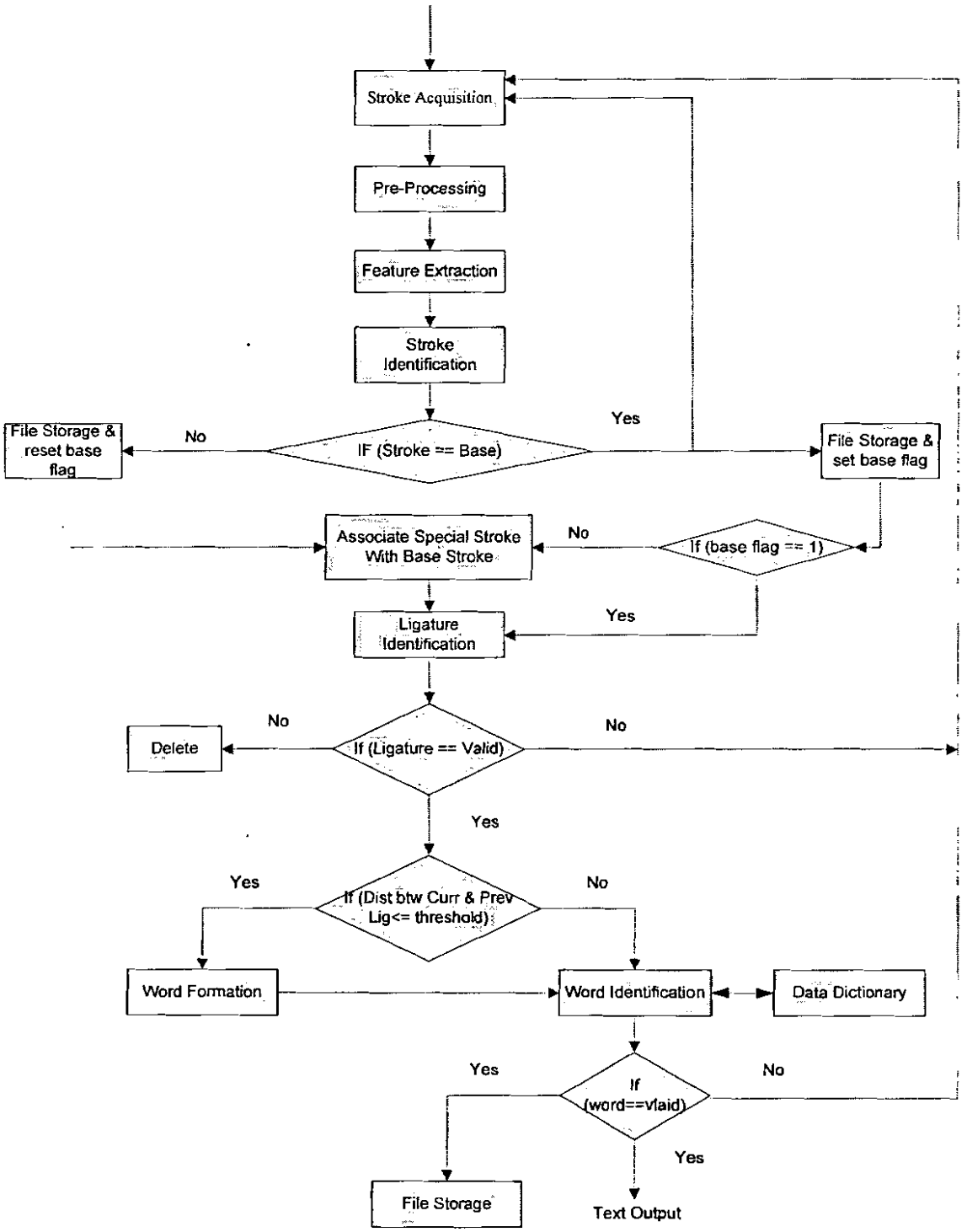


Figure 3.2: Stages of Urdu Character Recognition

3.5 Proposed 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 chapter2. The modules include:

3.5.1 Acquisition:

The processing was done on strokes obtained using digitizing tablet, the Intuos Wacom board. The tablet resolution is 2540 lpi (lines per inch) and pressure level 1024. The data capture application screen was set to the size of the monitor to standardize the input space and to smooth the data samples. The data is collected in real time. The standard data is a stream of {x, y} coordinates.

3.5.2 Preprocessing:

The data thus obtained often contains irregularity such as the hooks and erratic handwriting generated by inexperienced users. Hooks occur due to the inaccuracies during pen up and pen down while placing the stylus on, or lifting it off the tablet.

Figure 3.3: ج written by inexperienced writer and لر containing hooks in the beginning and end

1. Chain Coding:

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.

2. Dehooking:

Hooks are very common artifacts found at the ends of the strokes. They are generated during fast writing, 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.

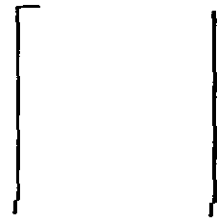


Figure 3.4: Alif before and after Dehooking

3. Smoothing:

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 e.g. the Urdu letter Laam gave the following chain codes before and after smoothing:

Before Smoothing:

" 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, 7, 7, 6, 6, 6, 6, 6, 6, 6, 6, 6, 5, 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 Smoothing:

" 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, 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, 2, 2, "

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

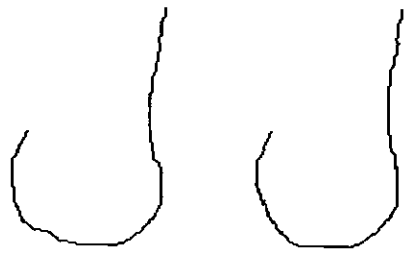


Figure 3.5: ل before and after smoothing

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 ligatures successfully.

3.5.3 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 features 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 ع feature which is selected on the presence of ع in any ligature. These features were very helpful in uniquely distinguishing the ligatures.

Features for Base Stroke:

The feature vector prepared for recognizing the base strokes includes the following features:-

1. **Start Vertical:**
   This feature was selected when the ligature was a straight vertical in the beginning e.g.ا , ط, ل.

2. **End Vertical:**
   This feature is selected when the ligature is straight vertical in the end e.g. م .

3. **Horizontal R2L:**
   If while writing the ligature the pen movement is from right to left horizontally then a bit is set for this feature vector e.g. in ب, ف.

4. **Horizontal L2R:**
   While writing the ligature, if the pen movement is from left to right horizontally then this feature is set in the feature vector e.g. ج, ے.

5. **Hedge:**
   In Urdu characters like, ن, س, ق, ص, ی a semi circle sort of shape is present in them which we call curve. For such characters we have selected a feature called the hedge.

6. **Curve R2L:**
   The direction of writing of these curves varies from right to left and also from left to right. Therefore, Curve R2L has been set for characters whose writing direction is right to left e.g. ن, ل.

7. **Curve L2R:**
   If the curve direction of the character is from left to right then this feature is set for them e.g. ج and ع .

8. **Loop Flag:**
   Loops are very common features of Urdu handwriting. They are present in characters such as م, و, ط, ف and ق. Whenever the recognition engine finds a loop it selects this feature for that character or ligature.

9. **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 س and سر as shown in the figure below.

Figure 3.6: Cusp in character س

10. **Intersection:**
    When ever an intersection is encountered in a stroke this feature is selected for that stroke e.g. these are present in س, ط, فل etc.

11. **Ray:**
    This feature is selected for the character ray of Urdu alphabet. If any ligature is a combination of ray then this feature is also selected for that particular ligature e.g. خر, خڑ, سر خز, سڑ,سز etc.

12. **End Up Vertical:**
    This feature is selected for the characters having a vertical end in the upward direction e.g. با, جا , صا , سا, ما, فا, غا, ظا etc.

13. **Loop Up:**
    In order to differentiate the loop in as ما, م, ف and ق, this feature was identified and selected for ما. The writing direction of the loop in ما is from down to up as shown in figure below.

Figure 3.7: Writing direction of loop of ما

14. Seen Bit:

This feature was selected if character seen is detected in any ligature e.g. بس, عس, سس, جش, جس, بش etc.

15. Aien Bit:

This feature was selected if character ع is detected in any ligature e.g. صع, بغ مع لع, كع, , فع, صغ, etc.

16. Hay Bit:

The presence of hay in any ligature is shown by the selection of this bit in the feature vector e.g. چچ , گح , ضح, سح, خح, حح etc.

17. Dal Bit:

If the recognition engine detects a د in the ligature written it selects this feature for it e.g. طڈ,طد شڈ,شد سڈ,سد خڈ,خد حڈ,حد etc.

18. Double Loop:

This feature is selected for characters and ligatures which have two loops in them e.g. ligatures like فع, مع, فق, قص etc have this feature selected in their feature vector.

Figure 3.8: double loop ligatures

19. Tuan Bit:

This feature is selected on the presence of ط in any ligature e.g. شط,سط , بط,چط, جط etc.

20. Gol Hay:

All the gol hay ligatures have this feature selected for them e.g. یہ , لہ, عہ, مہ, فہ etc. The shape of gol hay ligatures is given below

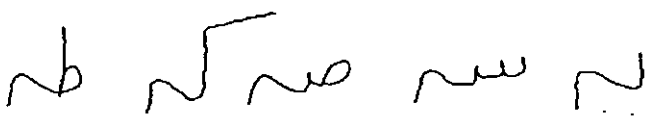


Figure 3.9: Gol Hay Ligatures

Features for Secondary Stroke:

The feature vector prepared for the secondary strokes includes the following features:-

1. **Dot:**

   If the secondary stroke has a length less then or equal to the threshold specified for a dot it is taken as a dot provided it is with in the boundaries of the base stroke. If a dot is encountered this feature is selected.

2. **Madaa:**

   If the stroke written is a secondary stroke and it fulfills the criteria set for the madaa stroke. This feature is selected then.

3. **Diagonal:**

   This secondary stroke feature is selected for the diagonal stroke occurring in ک and گ. If any base ligature has a diagonal over it this feature is selected.

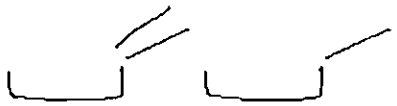


Figure 3.10: The diagonal stroke over ک and گ

4. **Hay:**

   This feature is selected if the secondary stroke called the hay ہ is encountered e.g. in ہا, ہٹ.

Figure 3.11: The hay stroke

5. **Hamzaa:**

   Hamzaa is a secondary stroke which is often present over the base strokes. If the secondary stroke is Hamzaa then this feature is selected.

6. **Chooti Tuan:**

   This secondary stroke is present over the base ligatures. If the loop follows a vertical line then the stroke is a ط. In this case this feature will be selected as a feature vector.

Figure 3.12: The ط over the base stroke kashti.

3.5.4 Stroke Identification:

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

1. The identification of the base stroke and
2. The identification of the secondary stroke.

Now, which phase works when? 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 the back-propagation neural network. The neural network then processes the inputs and generates the output. This output then helps us to classify the ligature.

3.5.5 Back-Propagation Neural Network:

A neural network is a powerful data modeling tool that is able to capture and represent complex input/output relationships. Back-Propagation neural network is a very popular type of neural network due to its application to different types of tasks. We have used this neural network because it is known to show great efficiency in character recognition applications especially those dealing with large data.

The Back-Propagation neural network consists of at least three layers; the input layer, hidden layer and an output layer. With Back-Propagation networks, learning occurs during a training phase in which each input pattern in a training set is applied to the input units and then propagated forward. The pattern of activation arriving at the output layer is then compared with the correct (associated) output pattern to calculate an error signal. The error signal for each such target output pattern is then back propagated from the outputs to the inputs in order to appropriately adjust the weights in each layer of the network. After a Back-Propagation network has learned the correct classification for a set of inputs, it can be tested on a second set of inputs to see how well it classifies untrained patterns. [3]

Back propagation Algorithm:

The procedure of training feed forward neural networks using the back propagation algorithm [20] is described as follows:
Assume there are m input units, n hidden units, and p output units.

1. Apply the input vector, Xp = ($Xp1$, $Xp2$, $Xp3$, ... , XpN)t to the input units .
2. Calculate the net- input values to the hidden layer units:

$$net^{h}_{pj}=(\sum_{i=1}^{N} W^{h}_{ji} \bullet X_{pi})$$

3. Calculate the outputs from the hidden layer:

$$i_{pj} = f^{h}{}_{j}(net^{h}{}_{pj})$$

4. Move to the output layer. Calculate the net-input values to each unit:

$$\text{net}^{\circ}\text{pk} = \left(\sum_{j=1}^{L} W^{\circ}\text{kj} \bullet i_{\text{pj}}\right)$$

5. Calculate the outputs:

$$O_{pk} = f^{O}{}_{j}(net^{O}{}_{pk})$$

6. Calculate the error terms for the output units:

$$\delta^{\circ}\text{pk} = (Y_{\text{pk}} - O_{\text{pk}}) \bullet f^{\circ'}\text{k}(\text{net}^{\circ}\text{pk})$$

$$\textit{Where,}\ f^{\circ'}\text{k}(\text{net}^{\circ}\text{pk}) = f^{\circ}\text{k}(\text{net}^{\circ}\text{pk}) \bullet (1 - f^{\circ}\text{k}(\text{net}^{\circ}\text{pk}))$$

7. Calculate the error terms for the hidden units

$$\delta^{h}{}_{pj} = f^{h'}{}_{j}(net^{h}{}_{pj}) \bullet \left(\sum_{K=1}^{M} \delta^{o}{}_{pk} \bullet W^{o}{}_{kj}\right)$$

Notice that the error terms on the hidden units are calculated *before* the connection weights to the output-layer units have been updated.

8. Update weights on the output layer

$$W^{\circ}\text{kj}(t+1) = W^{\circ}\text{kj}(t) + (\eta \bullet \delta^{\circ}\text{pk} \bullet i_{\text{pj}})$$

9. Update weights on the Hidden layer

$$W^{h}{}_{ji}(t+1) = W^{h}{}_{ji}(t) + (\eta \bullet \delta^{h}{}_{pj} \bullet X_{i})$$

Notations:

X_{pi} : net input to the i^{th} input unit
net^{h}_{pj} : net input to the j^{th} hidden unit
W^{h}_{ji} : weight on the connection from the i^{th} input unit to j^{th} hidden unit .
I_{pj} : net input to the j^{th} hidden unit
net ° pk : net input to the k^{th} output unit
W ° kj : weight on the connection from the j^{th} hidden unit to k^{th} output unit .
O_{pk} : actual output for the k^{th} output unit .
Y_{pk} : desired output for the k^{th} output unit .
F : (Sigmoid) activation function
f ' : derivative of activation function
δ °pk : signal error term for the k^{th} output unit.
δ ^{h} pj : signal error term for the th hidden unit.
h : learning rate

T.H-4475

Back-Propagation NN for Base Ligatures:

Our Back-Propagation network for the base ligature recognition consists of twenty inputs, sixty nine hidden nodes and One fifty three output neurons. The learning rate for this has been set to 0.6F.

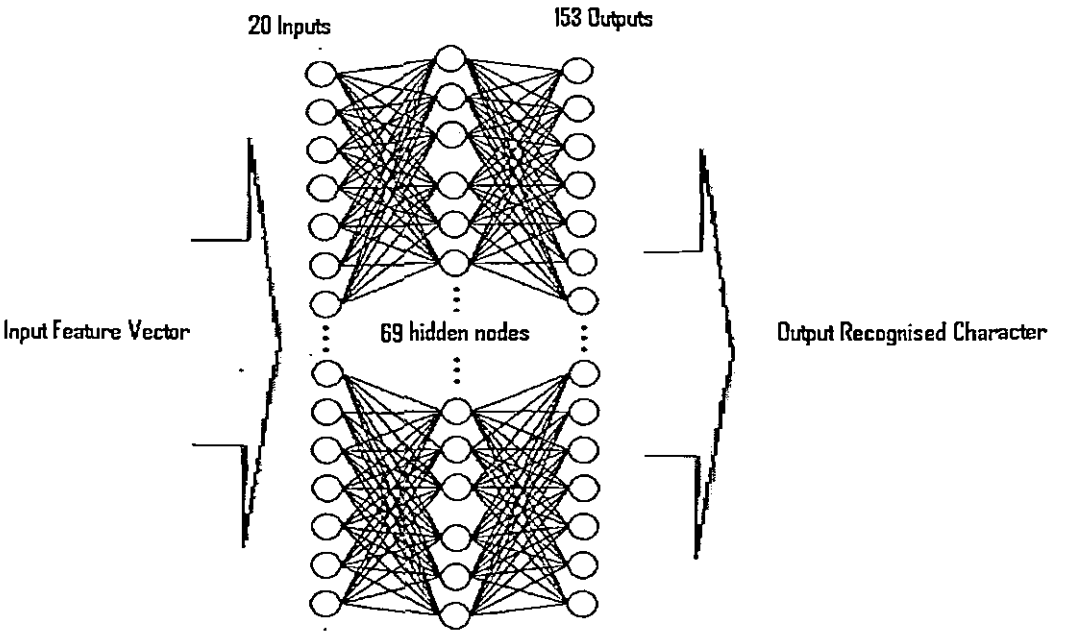


Figure 3.13: The Back-propagation Neural Network used in this research

Back-Propagation NN for Diacritics:

The Back-Propagation network for the secondary strokes recognition consists of five inputs, five outputs and ten hidden nodes. The learning rate for this has been set to 0.5F.

The number of hidden nodes and the learning rate were set through testing various values.

3.5.6 Ligature Validation:

In this stage, the ligature written is check for its validity. If the ligature is valid then the next ligature is considered. Otherwise, the ligature is removed to be rewritten.

3.5.7 Ligature Combination/ Word Formation:

In this stage the difference between ligatures is considered. If the difference between the previous ligature and the current ligature is less then the threshold which is necessary for word formation then these ligatures form one word. When the difference between successive ligatures increases then the new ligature is taken up as another word. In this way a series of words can be written. For example ا is one ligature and ک is another ligature. When we combine these two, the word کا is formed.

3.5.8 Intelligent Word Identification:

Once the words are written they are checked for validity. The word validity check means that whether the word written is a valid word or not. This stage makes use of the

dictionary to identify that the word written is valid or not. If the dictionary does not contain the word then the word is discarded. For example آتا is a valid word but آچا is not a valid word. Therefore, the dictionary will not identify it and discard it.

The dictionary comparison is based on the words Unicode comparison. If the Unicode of the word written is present in the dictionary then the word is a valid otherwise it is considered an invalid word and removed from the interface. An invalid word is also not written in the file.

3.5.9 Output:

The output is the Urdu Text in the interface's text area and in a word file. Only the valid words 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. The writing is done by sending Unicode to the file through the program.

آپ کام کرتے ہیں.

Figure 3.14: The text in file.

3.6 Conclusion:

In this chapter we have discussed the strategies used in this thesis. We have used a segmentation free approach for recognition. Using this approach we extracted a separate feature vector for the base strokes and the secondary strokes. A feature vector of 20 features was prepared for the base strokes. A feature vector of 6 features was prepared for the secondary strokes. The classification was performed using feed forward back propagation neural network. A total of 864 ligatures were trained and tested successfully. The Recognition rate of base ligatures was 93% and of the secondary strokes was 98%. The recognized ligatures as well as words and sentences were written in word file compatible with the Unicode standards.

Chapter 4
EXPERIMENTAL RESULTS

4.1 Introduction:

This work is an initial step in the field of Online Urdu Character Recognition. This work is by no means a concluding contribution. The work was initiated with the spirit of contributing to the national development, so that the national language can keep up its pace with modern times.

As previously discussed Urdu is a cursive language. We adopted a different approach by using the segmentation free approach because it has been observed that many recognition errors occur because of segmentation errors. 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 17000 ligatures. As this is quite a large number for the very limited time we had for our research. 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 and three ligature words. Due to extreme time constraints the three ligature words were only considered for the very common occurrences.

Once this was decided, another problem was the availability of the data base of the most frequently used ligatures. The list of the one, two and three ligature words was obtained from Ahmed Mirza Jamils computerized book of more than 17,000 ligatures. The second problem was choosing the frequent ligatures out of these. For this help was taken from two most common sources of Urdu ligatures. The First source was the Urdu newspapers namely "The daily Nawa-i-Waqt" and "The Jang". The second was the Urdu websites " http://www.bbc.co.uk/urdu/". As a result approximately five hundred ligatures including those ligatures that vary with different secondary strokes and their different positions were chosen for recognition. These 500 ligatures form more than 1000 words which was quite an enough target for our preliminary research. A list of the ligatures is provided in the Appendix 2.

4.3 Urdu Data Dictionary:

For the valid Urdu word verification a database of valid Urdu words was required. This database was obtained from the "http://www.urduweb.org"under the general public license. They provided a database of 33651 words under the general public license agreement. This database was then modified by keeping only the one, two and three very commonly occurring ligature words.

4.4 Ligature Testing and Results:

All the ligatures were tested before proceeding to the word formation stage. These ligatures were tested by six university students. Some of the problems encountered during this process were the difficulty of writing on the digital board by the inexperienced writers. The Recognition rate of base ligatures was 93% and of the secondary strokes was 98%.

4.5 Experimental Results:

4.5.1 Recognition of Ligatures:

Screen shots of recognition of one, two and three character ligatures are given below.

- **Recognition of س and ش:**

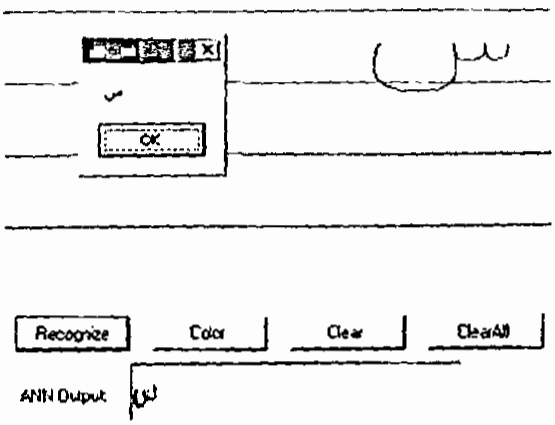


Figure 4.1: س recognized

If we place three dots over س it becomes ش which has also been successfully recognized.

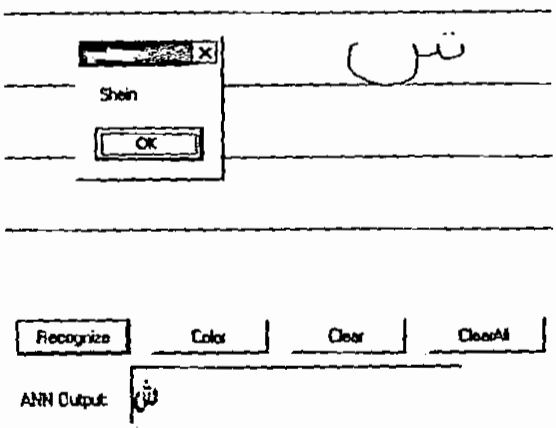


Figure 4.2: ش recognized

- **Recognition of Base Stroke Kashti and ک,ٹ:**

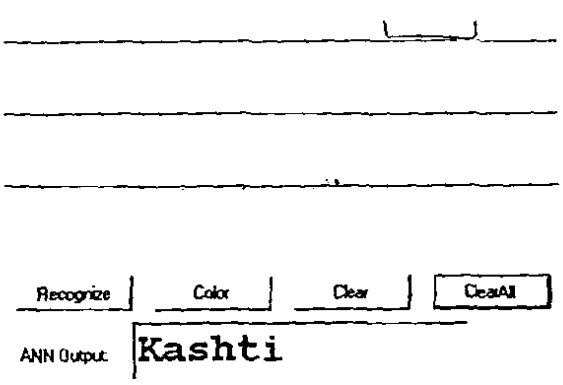


Figure 4.3: Base ligature kashti recognized

Placing a secondary stroke over or under the base ligature changes its identity. Placing the choti tuan over the kashti makes it aٹ and placing diagonal makes it ک.

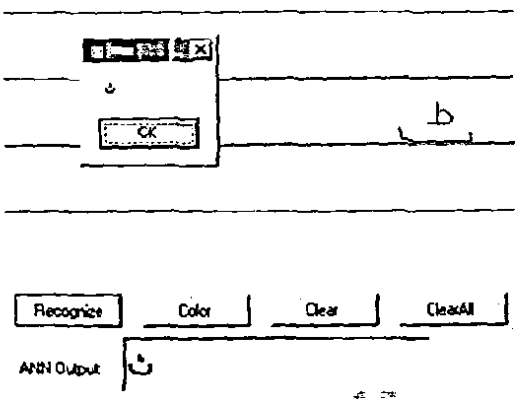


Figure 4.4a: ٹ recognized

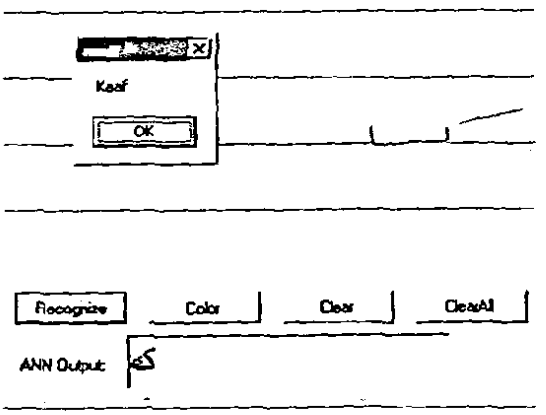


Figure 4.4b: ک recognized

- **Recognition of Base Stroke لر and گر :**

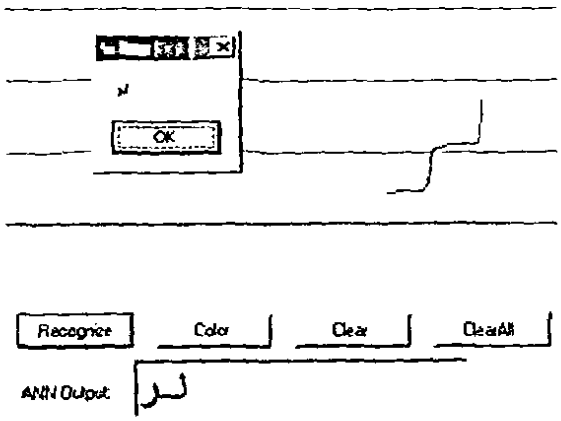


Figure 4.5: لر recognized

When the لر is followed by the secondary stroke the diagonal it becomes گر.

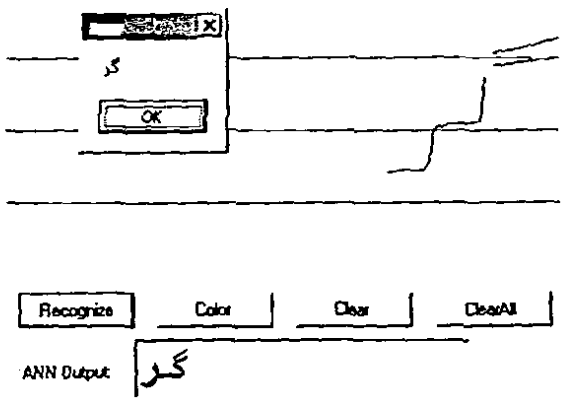


Figure 4.6: گر recognized

- **Recognition of تھا:**

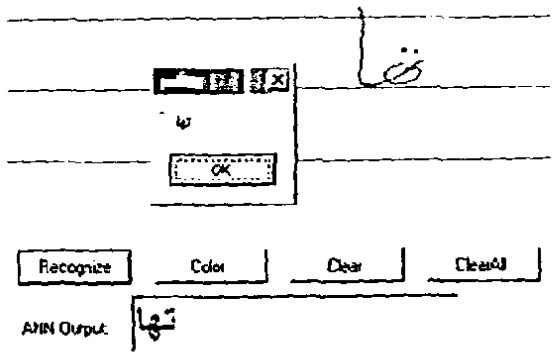


Figure 4.7: تھا recognized

- **Recognition of لٹا and کیا:**

Placing a secondary stroke over or under the base ligature changes its identity. Placing the choti tuan makes it a لٹا and placing diagonal over the base stroke and dots under it makes it کیا

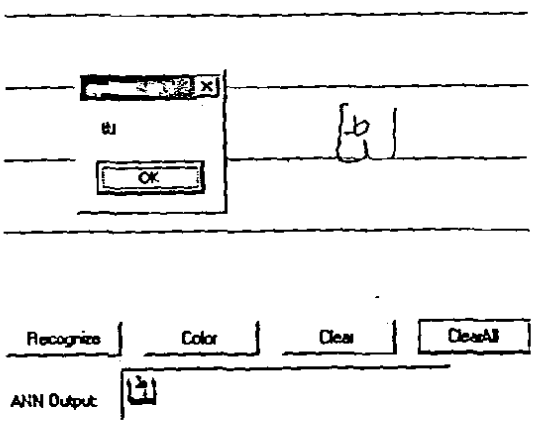


Figure 4.8a: لٹا recognized

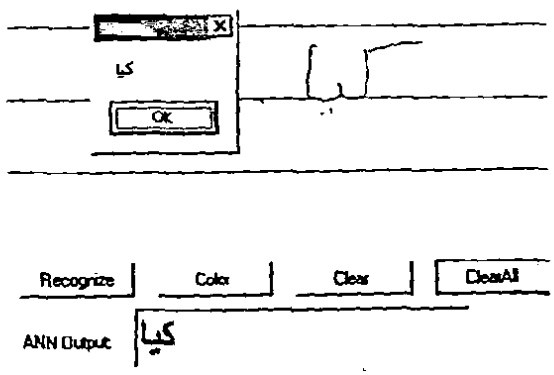


Figure 4.8b: کیا recognized

- **Recognition of بٹن and ہیں:**

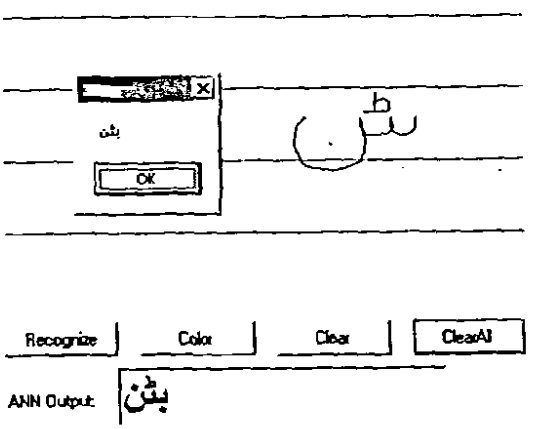


Figure 4.9a: بٹن recognized

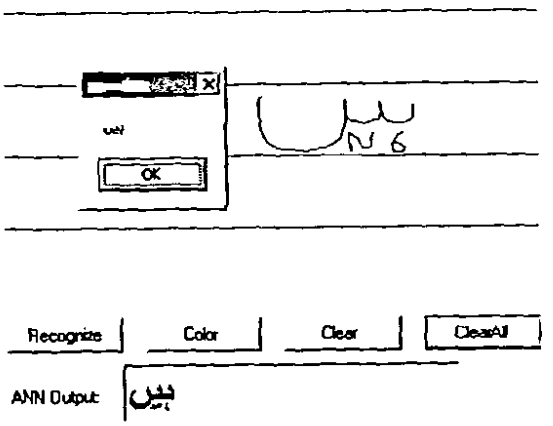


Figure 4.9b: ہیں recognized

4.5.2 Recognition of Words:

Screen shots of the valid words formed from the combination of these ligatures are given below.

- **Recognition of با ت:**

  The word با ت which is formed of two valid ligatures با and ت was recognized successfully.

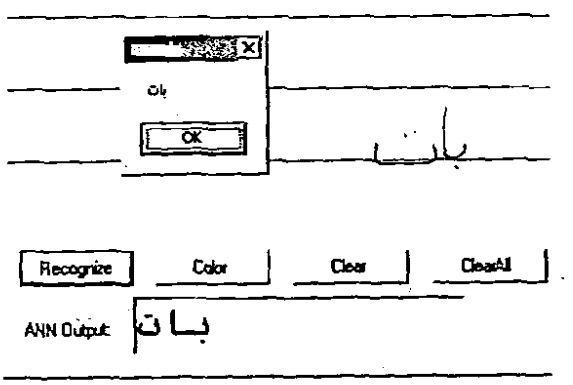


Figure 4.10: با ت recognized

- **Recognition of پھول :**

The word پھول which is formed of two valid ligatures پھو and ل was recognized successfully

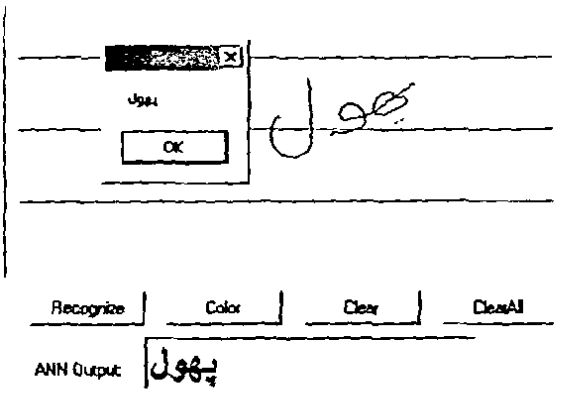


Figure 4.11: پھول recognized

4.5.3 Recognition of Sentence formed from valid Words:

Screen shots of the sentence formed by valid words are given below.

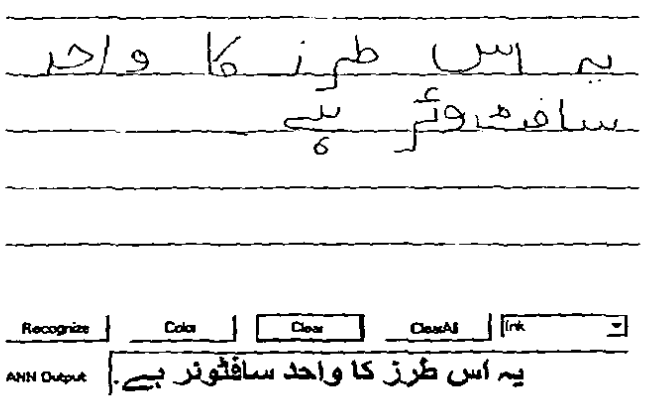


Figure 4.12: Sentence recognized

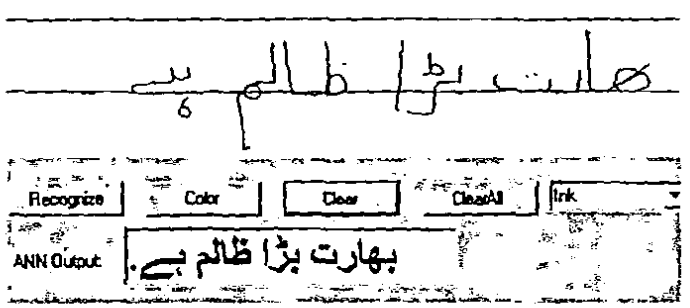


Figure 4.13: Sentence recognized

4.5.4 Recognition of Invalid Samples:

Our software has also successfully identified the invalid samples. For e.g. samples such as invalid secondary strokes over valid characters/ ligatures.

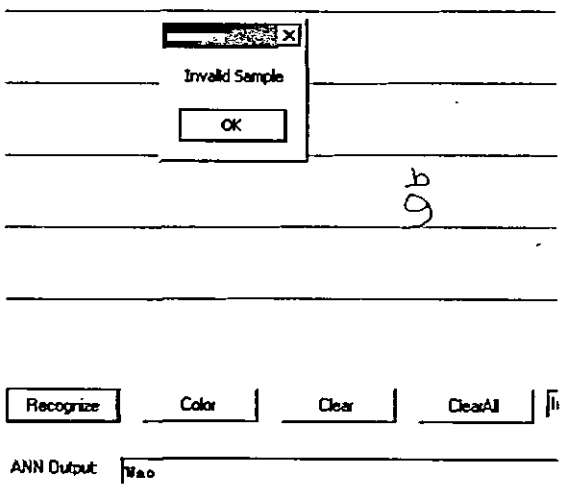


Figure 4.9: ؤ with invalid Secondary stroke identified

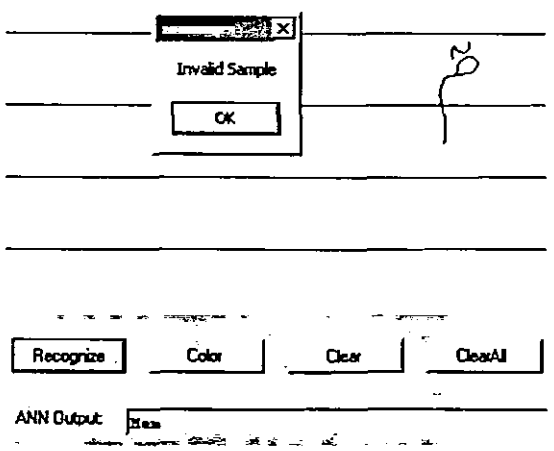


Figure 4.10: Invalid ہ identified

Test results of some of the difficult ligatures are given in the table below.

No	Ligature	Total Samples	Recognition Rate w.r.t. samples in %
1.	ب class	18	89.3
2.	س class	18	90.6
3.	ص class	18	87.3
4.	ط class	18	93
5.	یا	18	96.8
6.	طا	18	86.5
7.	طب	18	87.3
8.	فب	18	87.3
9.	صح	16	83.5
10.	حح	16	95.5
11.	جد	16	95.5
12.	قص	16	95.5
13.	فق	16	95.5
14.	مق	16	95.5
15.	مع	16	85.7
16.	عے	16	94.5
17.	طو	16	94.5
18.	سہ	16	91
19.	تھو	16	87.5
20.	ہیں	16	96.5

Table 1: Recognition Rate of base ligatures w.r.t. samples

Special Ligature Identification Results:

The recognition rate of these ligatures was very good.

No	Ligature	Recognition Rate
1.	٠	100 %
2.	~	85 %
3.	/	100 %
4.	،	95 %
5.	ط	95 %
6.	ء	95 %

Table 2: Recognition Rate of diacritics w.r.t. samples

All the 864 ligatures were trained and tested during the software development by two writers. The final testing was performed on some selected ligatures some of which are mentioned in the table above.

4.6 Conclusion:

The testing and training of the handwritten ligatures, words and sentences has given quite successful results. The issue that remains is the difficulty in writing on the tablet with a pen. The use of the tablet pen has been found a little difficult for the beginners. This is not a big issue, as the writers learn this in few trials.

The list of all the trained ligatures and some of the recognized words is provided in the Appendix 1 and Appendix 2 respectively.

Chapter 5
CONCLUSION & DISCUSSION

5.1 Introduction:

This work is an initial step in the field of Online Urdu Character Recognition. This work is by no means a concluding contribution. The work was initiated with the spirit of contributing to the national development, so that the national language can keep up its pace with modern times.

As previously discussed Urdu is a cursive language. We adopted a different recognition technique, by using the segmentation free approach because it has been observed that many recognition errors are due to the wrong segmentation. 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.

5.2 Results Discussion:

Before the testing and the training of the neural network, comes the designing of the feature vector. Deducing unique feature out of every handwritten may seem quite easy visually, but the implementation shows that the distinguishing feature that is was quite avid visually is not so when it comes to making the logic for it. We faced this problem so many times.

For example ا alif and ر ray have a visually different shape i.e. ا is long straight vertical and ray is short vertical in the beginning and then there is a short horizontal to the left. If the ا rather then being straight is written tilted towards the left or right it is recognized as ا because the chain codes generated for both the characters become same. ا in this case might have chain codes 6,5 or 6,7. There is no problem with 6,7 but 6,5 is a problem as ray also generates the same chain code if the short horizontal of ray not written straight but a little tilted in the South direction as shown in figure 1.

Figure 1: ا and ر written tilted.

Similarly, ص and ق are again quite distinct visually but when one sits to program the same quite evident difference you are again stuck. We see quite clearly that the loop for the ق is with in the curve while that for the ص is on the right hand side of the curve. For this the loop end should be properly identified. If the loop is properly written that is it a closed well defined loop then there is no problem but if the writer left the loop a little open then the loop may be identified as ح.

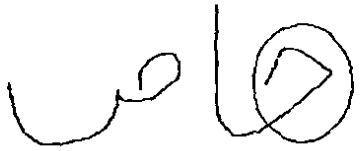


Figure 2: Problems arising due to open loop of ص which is similar to ح.

Another tough recognition point was the difference between قص and فق.

Figure 3: قص and فق.

If the loops are written properly and the loop ends also identified then the next problem was how to differentiate the ص loop and the ق loop as highlighted in figure 4.

Figure 4: The distinguishing feature of the قص and فق loop.

All these problems have been handled in our research. Therefore, even if the writer writes the characters a little tilted or improper the software would recognize them.

5.3 Constraints:

The constraints for the system are that the base stroke should be written before the secondary stroke. For the two character ligature expecting a secondary stroke, the secondary stroke for the first character should be written first and for the second character should be written second.

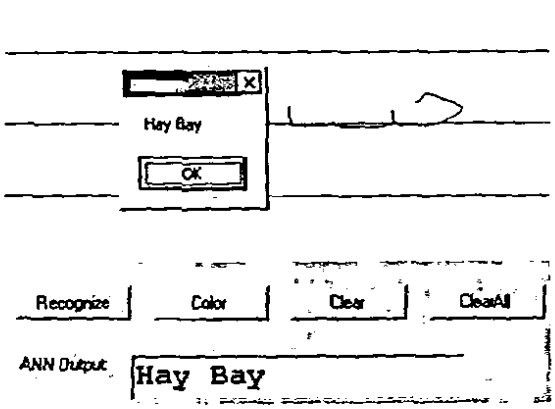


Figure 5a: Base Stroke

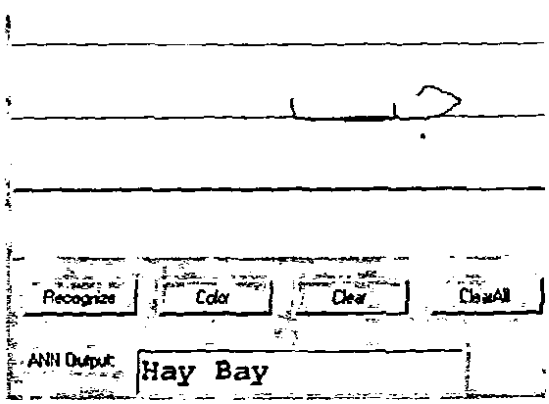


Figure 5b: secondary stroke for first character

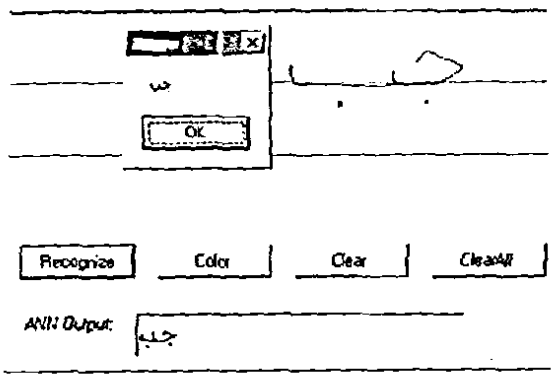


Figure 5c: secondary stroke for second character

Further, loops should be proper. Open loops greater than the threshold are not entertained. Similarly ligatures that become so close that they fulfill the loop threshold may accidentally be taken as loops.

5.4 Contribution:

In the field of online Urdu character recognition our contributions are:

- The recognition of 500 two-three character ligatures.
- The development o the software that successfully recognizes these ligatures.
- Recognition of the words formed with these ligatures.
- Filing of the valid words into a word file for further modification.

5.5 Future Directions:

As our research and implementation was an initial step. Therefore, there is a lot of scope for future enhancement.

- Implementation of other pre-processing techniques such as the RTS techniques.
- Enhancement in the number of ligatures which we think is a continuous area of research.
- Recognition of additional secondary strokes such as the shad, zeer, zabar and paish.

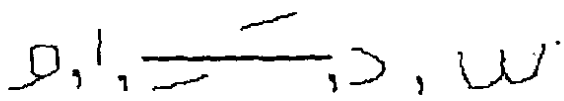


Figure 6: Additional secondary strokes

- Recognition of Urdu numerals.

5.6 Conclusion:

Urdu online character recognition is a continuous are of research and improvement. The more the development in this field the better will be the final product. The aim should be the development of a standard commercial product capable of incorporating more and more handwriting styles, hence giving access to public at large.

REFERENCES:

1. http://www.ethnologue.com/show_language.asp?code=urd.

2. http://www.omniglot.com/writing/urdu.htm.

3. http://www2.psy.uq.edu.au/~brainwav/Manual/BackProp.html

4. http://std.dkuug.dk/JTC1/SC2/WG2/docs/n2413-3.pdf.

5. Mohammad S. Khorsheed, William F. Clocksin, "Structural features of cursive Arabic script", proceeding of 10th British Vision, Conference, University of Nottingham, UK, September-1999.

6. M S Khorsheed,"Off-Line Arabic Character Recognition A Review".

7. Mohammad S. Khorsheed,"Automatic recognition of words in Arabic manuscripts", PhD Dissertation, Churchill College, University of Cambridge, June 2000.

8. H. Bunke, P. Wang, "Handbook of character recognition and document image analysis", World Scientific, 2000.

9. Syed Afaq Husain and Syed. Hassan Amin, A Multi-tier Holistic approach for Urdu Nastaliq Recognition, proceeding of the INMIC 2002.

10. Zahra A Shah and Farah Saleem. Ligature Based Optical Character Recognition of Urdu, Nastaleeq Font, proceeding of the INMIC 2002.

11. Sutat Sae-Tang Ithipan Methaste. Thai Online Handwritten Character Recognition Using Windowing Backpropagation Neural Networks, Information Research and Development Division, National Electronics and Computer Technology Center, National Science and Technology Development Agency, Rachathewi, Bangkok 10400, Thailand.

12. VKazushi Ishigaki VHiroshi Tanaka VNaomi Iwayama. Interactive Character Recognition technology for Pen-based Computers.

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

14. A.Amin, G. Masini and J.P. Haton, Recognition of Handwritten Arabic Words and Sentences, proceeding of the 7th Int.Conf on Pattern Recognition, Montreal, 1984, 1055-1057.

15. Eran Aharonson, Speech Technology Magazine, issue July 1999, As with Speech, Online Handwriting Recognition Enables PCs to Understand Natural Human Input.

16. I.Guyon, J.Bromley, N.Matic, etc, "A neural network system for recognizing on-line handwriting", Models of Neural network, Springer Verlag, 1996.

17. Sarmad Hussain and Muhammad Afzal, "Urdu Computing Standards", Urdu Zabta Takhti (UZT) 1.01 - WG2 N2413-3 / SC2 N3589-3

18. 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.

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

20. Freeman, J. A. &, Skapura, "Back propagation. Neural Networks Algorithm Applications and Programming Techniques", D. M. (1992), 89-125.

List of recognized one character ligatures						
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					

List of recognized two character ligatures						
لا						
با	پا	تا	ٹا	ثا		
جا	چا	حا	خا			
صا	ضا					
سا	شا					
ما						
فا	قا	کا	گا			
عا	غا					
طا	ظا					
با						
با						
نا						
ها	نها	بها	پها	نها	ٹها	
ببـ	ببـ	بتـ	بٹـ	بثـ		
پبـ	پبـ	پتـ	پٹـ	پثـ		
تبـ	تبـ	تتـ	تٹـ	تثـ		
ٹبـ	ٹبـ	ٹتـ	ٹٹـ	ٹثـ		
ٹبـ	ٹبـ	ٹتـ	ٹٹـ	ٹثـ		
جبـ	جبـ	جتـ	جٹـ	جثـ		
چبـ	چبـ	چتـ	چٹـ	چثـ		

حب	حپ	حت	حٹ	حٹ		
خب	خپ	خت	خٹ	خٹ		
سب	سٹ	سپ	ست	سٹ		
شب	شٹ	شپ	شت	شٹ		
طب	طپ	طت	طٹ	طٹ		
ظب	ظپ	ظت	ظٹ	ظٹ		
عب	عپ	عت	عٹ	عٹ		
غب	غپ	غت	غٹ	غٹ		
فب	فپ	فت	فٹ	فٹ		
قب	قپ	قت	قٹ	قٹ		
کب	کپ	کت	کٹ	کٹ		
گب	گپ	گت	گٹ	گٹ		
لب	لپ	لت	لٹ	لٹ		
مب	مپ	مت	مٹ	مٹ		
نب	نپ	نت	نٹ	نٹ		
بب	بپ	بت	بٹ	بٹ		
یب	یپ	یت	یٹ	یٹ		
نب	نٹ	نپ	نت	نٹ		
لک	لگ	ہک	ہگ	کک	کگ	گگ
نک	یک	پک	تک	ٹک	ٹگ	یگ
نگ	یگ	پگ	تگ	ٹگ	ٹگ	یگ
بج	بج	بچ	بخ			
پج	پج	پچ	پخ			
تج	نج	تچ	تخ			
ٹج	ٹج	ٹچ	ٹخ			
ٹج	ٹج	ٹچ	ٹخ			
جج	جج	جچ	جخ			
چج	چج	چچ	چخ			
حج	حج	حچ	حخ			
خج	خج	خچ	خخ			
سج	سج	سچ	سخ			
شج	شج	شچ	شخ			
صج	صج	صچ	صخ			
ضج	ضج	ضچ	ضخ			
کج	کج	کچ	کخ			
گج	گج	گچ	گخ			
لج	لج	لچ	لخ			
مج	مج	مچ	مخ			
نج	نج	نچ	نخ			
بج	بج	بچ	بخ			
یج	یج	یچ	یخ			
نج	نج	نچ	نخ			
بد	بڈ	بڈ				
پد	پڈ	پڈ				

تذ	تڈ	تڈ				
ٹذ	ٹڈ	ٹڈ				
ثذ	ثڈ	ثڈ				
جذ	جڈ	جڈ				
چذ	چڈ	چڈ				
حذ	حڈ	حڈ				
خذ	خڈ	خڈ				
سذ	سڈ	سڈ				
شذ	شڈ	شڈ				
طذ	طڈ	طڈ				
ظذ	ظڈ	ظڈ				
عذ	عڈ	عڈ				
غذ	غڈ	غڈ				
فذ	فڈ	فڈ				
قذ	قڈ	قڈ				
لذ	لڈ	لڈ				
کذ	کڈ	کڈ				
گذ	گڈ	گڈ				
مذ	مڈ	مڈ				
نذ	نڈ	نڈ				
بذ	بڈ	بڈ				
پذ	پڈ	پڈ				
ندّ	نڈّ	نڈّ				
بر	بڑ	بڑ				
پر	پڑ	پڑ				
تر	تڑ	تڑ				
ٹر	ٹڑ	ٹڑ				
ثر	ثڑ	ثڑ				
جر	جڑ	جڑ				
چر	چڑ	چڑ				
حر	حڑ	حڑ				
خر	خڑ	خڑ				
سر	سڑ	سڑ				
شر	شڑ	شڑ				
صذ	صڈ	صڈ				
ضذ	ضڈ	ضڈ				
طر	طڑ	طڑ				
ظر	ظڑ	ظڑ				
عر	عڑ	عڑ				
غر	غڑ	غڑ				
کر	کڑ	کڑ				
گر	گڑ	گڑ				
لر	لڑ	لڑ				

مز	مڑ	مز				
نز	نڑ	نز				
بز	بڑ	بز				
یز	یڑ	یز				
نز	نڑ	نز				
بس	پس	تس	ٹس	ٹس		
بش	پش	تش	ٹش	ٹش		
جس	چس	حس	خس			
جش	چش	حش	خش			
سس	سش	شش				
عس	غس					
عش	غش					
فس	قس					
فش	قش					
کس	گس					
کش	گش					
لس	لش					
مس	مش					
نس	نش					
یس	یش					
بس	بش					
نس	نش					
بص	پص	تص	ٹص			
بض	پض	تض	ٹض			
فص	فض					
قص	قض					
لص	لض					
نص	نض					
یص	یض					
بص	بض					
نص	نض					
سط	شط					
جط	چط	حط	خط			
نط	بط	پط	تط	ٹط	ٹط	
فط	قط					
عط	غط					
بط						
یط						
نط						
سظ	شظ					
جظ	چظ	حظ	خظ			
بظ	پظ	تظ	ٹظ	ٹظ		
فظ	قظ					
عظ	غظ					

بظ						
بظ						
نظ						
بع	بع	تع	ثع	ثع		
بغ	بغ	تغ	ثغ	ثغ		
صع	ضع					
صغ	ضغ					
فع	قع					
فغ	قغ					
كع	گع					
كغ	گغ					
لع	لغ					
مع	مغ					
نع	نغ					
بع	بغ					
بع	بغ					
نع	نغ					
بف	پف	تف	ثف	ثف		
سف	شف					
صف	ضف					
طف	ظف					
ڤف	قف					
كف	گف					
لف						
نف						
بف						
بف						
نف						
بق	پق	تق	ثق	ثق		
جق	چق	حق	خق			
سق	شق					
طق	ظق					
فق	قق					
كق	گق					
لق						
مق						
نق						
بق						
بق						
نق						
بل	پل	تل	ثل	ثل		
جل	چل	حل	خل			
سل	شل					
طل	ظل					

غل	غل					
قل	قل					
كل						
مل						
نل						
بل						
يل						
ںل						
بم	پم	تم	ثم	ثم		
جم	چم	حم	خم			
سم	شم					
صم	ضم					
طم	ظم					
عم	غم					
فم	قم					
كم	گم					
لم						
نم						
بم						
يم						
نم						
بن	پن	تن	ثن	ثن		
جن	چن	حن	خن			
سن	شن					
طن	ظن					
عن	غن					
فن	قن					
كن	گن					
لن						
من						
نن						
بن						
ين						
نن						
بو	پو	تو	ثو	ثو		
جو	چو	حو	خو			
سو	شو					
صو	ضو					
طو	ظو					
عو	غو					
فو	قو					
كو	گو					
لو						
مو						

نو						
بو						
یو						
نو						
بہ	پہ	تہ	ٹہ			
جہ	چہ	حہ	خہ			
سہ	شہ					
صہ	ضہ					
طہ	ظہ					
عہ	غہ					
فہ	قہ					
کہ	گہ					
لہ						
مہ						
نہ						
بہ						
یہ						
نہ						
بی	پی	تی	ٹی	ٹی		
جی	چی	حی	خی			
سی	شی					
صی	ضی					
طی	ظی					
عی	غی					
فی	قی					
لی						
می						
نی						
بی						
یی						
نی						
بے	پے	تے	ٹے	ٹے		
جے	چے	حے	خے			
سے	شے					
صے	ضے					
طے	ظے					
عے	غے					
فے	قے					
کے	گے					
لے						
مے						
نے						
بے						
نے						

ھے						
ھی						
ھو						

List of recognized three character ligatures						
بھے	پھے	تھے	ٹھے	نھے		
بھی	پھی	تھی	ٹھی	نھی		
بھو	پھو	تھو	ٹھو	نھو		
ہیں	ہیں	ہیں	ٹیں	ٹیں	نیں	نیں
ہیں	ہیں	ہیں	ٹین	ٹین	نین	نین
بتن	بٹن	پتن	پٹن			
میں	مین					
لبا	لپا	لتا	لٹا	لٹا	لنا	
لیا	کیا	گیا	کٹا	کٹا	کبا	
لبو	لپو	لتو	لٹو	لٹو	لنو	
لیو	کیو	گیو				

Total ligatures: 864 ligatures

APPENDIX 2

ا
آ
آو
آوٹ
آوں
آنی
آنے
اب
آب
ابا
آبا
آباواجداد
آبائی
اباجی
آباد
آبادگاری
آبادی
آبادیات
آبادیوں
اباسی
ابال
ابد
ابدا
آبدار
ابدال
آبدوز
آبدوزیں
ابدی
ابدیت
آبدیدہ
ابر
ابرار
ابرآلود
آبرو
آبرونے
ابروبارا
ں
ابل
ابھی
ابو

ابواب
ابوالاسود
ابوحاتم
ابوداود
ابوذر
ابوطالب
ابوعامر
ابوہریرہ
ابھار
ابھارتا
ابھارتے
ابھارتا
ابھارنے
ابھاریں
ابھارے
ابھاگن
ابھی
آپ
آپا
آپادھاپی
اپاہج
اپریل
آپی
آپے
آتا
اتار
اتارا
اتارتا
اتارتی
اتارتے
اتارنا
اتارنی
اتارنے
اتارو
اتاریں
اتر
اترا
اترائے
اتراتا
اتراتے
اتروانا

اترواپا
اتروں
اتریں
اتوار
آتی
آتے
اثاثہ
آثار
اثر
اثرات
اثرانداز
اثرپذیر
آج
آجا
اجارہ
اجازت
اجازتوں
اجازتی
اجاگر
اجالا
اجالوں
اجاڑ
اجاڑا
اجداد
اجر
اجرت
اجرک
اجروثواب
اجزا
اجل
اجڈ
اجڑ
اجڑا
اجڑنے
اچار
اچانک
اچاٹ
اچک
اچٹ
احادیث

احاطہ
احاطے
احد
احدی
احرام
احوال
اخذ
آخر
اخراج
اخراجات
آخرت
آخرکار
آخری
اخوت
ادا
آداب
ادارت
اداروں
ادارہ
اداریہ
ادارے
اداس
اداسی
اداکار
اداکاروں
اداکاری
ادب
ادبی
ادراتی
ادراک
ادرش
ادرک
ادرڑکا
ادریس
ادغام
ادلے
آدم
آدمی
ادوائن
ادوار

ادویات
آدھا
ادھار
ادھورا
آدھوں
آدھی
آدھے
ادیان
ادیب
آذادی
آذادیاں
آذان
اذان
اذانوں
اذاں
آذر
اذن
اذیت
آرا
آرائش
آرائی
آرائے
ارادوں
ارادہ
ارادی
ارادیت
ارادے
اراضی
آرام
ارباب
ارباز
اربوں
ارحم
اردگرد
اردلی
اردن
اردو
اردوداں
ارزاں
آرزو

آرزوئیں
آرزوئے
ارسال
آرسی
ارشاد
ارشادات
ارشد
ارض
ارضی
ارفع
ارکان
ارکیالوجی
ارم
ارمان
ارمانوں
ارماں
آرمی
ارواح
آرٹ
آرٹس
آری
آریا
آریای
آریوں
ارے
آرے
از
آزاد
آزادانہ
آزادی
آزادیاں
آزادیوں
آزار
آزاری
ازالہ
ازالے
ازاں
ازبر
ازحد
ازخود
ازدواجی

آزر
ازسر
ازسرنو
آزمائی
آزمایش
آزمائیں
آزمایے
آزمائے
آزمانا
آزمودہ
ازواج
اژدحام
اژدھا
اژدھے
اس
آس
آسائش
اساتذہ
اساس
اسامہ
آسامی
آسان
آسانی
اسد
آسرا
اسرایل
اسرار
اسم
اسود
آسودگی
آسودہ
اسوہ
اسی
اسے
اشارے
اشارات
اشاروں
اشارہ
اشارے
اشاعت

اٹھا	آواز	اندازی	امانت	اکڑوں	آفاقی	آشانہ
اٹھاو	آوازیں	اندازے	امت	آگ	آفت	اشد
اٹھائیں	اوباش	اندر	آمد	اگائی	افراد	اشرف
اٹھاتا	اوپر	اندرا	امداد	اگاتا	افراط	اشرفی
اٹھارواں	اور	اندراج	آمدن	اگالدان	افروز	اشک
اٹھارویں	اوراق	اندراجات	آمدنی	اگانے	آفریدی	آشوب
اٹھارہ	اورنج	اندرون	آمدورفت	اگاہ	آفرین	اشوک
اٹھارہواں	اوروں	اندرونی	آمدید	آگاہ	افزا	اصرار
اٹھارہویں	اوزار	اندروں	امر	آگاہی	افزائش	اصل
اٹھاسی	اوزان	اندھا	امرا	اگتا	آفس	اصول
اٹھان	اوس	اندھوں	امراض	اگر	افق	اصولوں
اٹھانا	اوسان	اندھی	آمرانہ	اگرچہ	افواج	اضافت
اٹھاتوے	اوسط	آندھی	امرود	آگرہ	افواہ	اضافوں
اٹھاون	اوصاف	اندھے	امروز	اگل	افواہیں	اضافہ
اٹھایا	اوقات	اندیش	آمریت	الاو	آقا	اضافی
اٹھائے	اوقاف	انرجی	امن	الائش	اقارب	اضافے
اٹی	اوکاڑہ	انس	امور	آلائش	اقامت	اطاعت
اٹے	اول	انگ	آموز	الاونس	اقامہ	اطالوی
آٹے	اولاد	انوار	آموزی	الاپ	اقدار	اطراف
اڈا	اولادیں	انواع	آموں	آلات	اقدام	اطوار
اڈریس	اولیا	انور	امڈ	الف	اقدس	اعانت
اڈوانس	اومان	انڈا	امی	الگ	اقرار	اعداد
اڈہ	اون	انڈوں	امین	الناس	اقربا	اعراب
اڑ	اونچ	انڈہ	آمین	الو	اقوال	اعرابی
آڑ	اوندھا	انڈیا	امیں	آلو	اقوام	اعراض
اڑا	اونٹ	انڈیز	ان	آلووں	اک	اعزاز
اڑائی	اونی	انڈیل	آن	الٹ	اکاونٹ	اعزازیہ
اڑائیں	اوٹ	انڈین	انا	الٹا	اکابر	اعوان
اڑانے	اوڑھا	انڈے	آنا	الہ	اکابرین	اعوذ
اڑاتا	اوڑھانے	آنہ	اناج	آلہ	آکاش	آغا
اڑائی	اوں	آنی	انار	الیاس	اکاون	آغاز
اڑان	اوہ	آنے	اناروں	آلے	اکتا	اغراض
اڑانا	اوہو	او	اناسی	آم	اکتاتا	اغوا
اڑتا	آویز	اونی	اناڑی	آمادگی	اکتاہٹ	آغوش
اڑتی	آویزاں	اونے	آنت	آمادہ	اکرام	آغوشی
اڑوانے	اویس	اوائل	انچ	امارت	اکرم	اف
اڑیل	آوے	اوابین	آنچ	امام	اکڑ	آفات
آڑے	آٹا	اواخر	انداز	امامت	آکڑ	افادیت
آں	اٹک	آوارگی	اندازوں	امامہ	اکڑا	آفاق
آہ	اٹل	آوارہ	اندازہ	امان	اکڑنے	آفاقہ
آہا						

برسے
برش
برطانوی
برف
برفانی
برق
برقرار
برقع
برقی
برکات
برکت
برکتوں
برگد
برگر
برما
بروئے
برڈ فلو
برہان
برہم
بری
بریانی
بریک
بریڈ
بزدل
بزدلانہ
بزدلی
بزرگ
بزرگان
بزرگانہ
بزرگوار
بزرگوں
بزرگی
بزم
بس
بک ڈپو
بل
بن
بو
بوئی
بوائی

بر
طرف
بر
گزیدہ
بر وقت
برا
برائی
برائے
برائے کرم
برابر
برات
براتیں
برادر
برادرانہ
برادری
برازیل
براق
برآمدہ
برانچ
برباد
بربادی
بربریت
برپا
برت
برتاو
برتر
برتن
برتیں
بردار
بردبار
برزخ
برس
برسا
برسانے
برسات
برساتیں
برساتا
برسانے
برسر الزام
برسوں
برسی

بد بو
بد
بوئیں
بد تر
بد حال
بد
حواس
بد خواہ
بد ظن
بد مزاج
بدمزگیاں
بد مزہ
بد نام
بدو
بدر
بدعا
بدعاوں
بدعائیں
بدعات
بدعت
بدل
بدلا
بدلاتا
بدلاتے
بدلتا
بدلنا
بدلو
بدلوا
بدلوں
بدلہ
بدلے
بدن
بدولت
بدھو
بذات
خانہ
بر
بٹن
بر
بچ
خاست
بچ
بر
داشت
بد
بد اماں

بالوں
بالی
بالیاں
بالیوں
بان
باندھا
باندھو
باندھی
باندھے
باندی
باندیاں
بانس
بانگ
بانو
بانوں
بانٹ
بانڈ
بانی
بانے
باوا
باور
باورچی
باولر
باون
باڈی
باڑ
باڑی
باڑے
بابر
بابیں
بایاں
بایں
بت
بت
خانہ
بٹن
بچ
بچ
بد
بد اماں

بارش
بارشوں
بارگاہ
بارڈر
بارھویں
بارہ
بارہویں
باری
باریک
بارے
باز
بازار
بازاروں
بازو
بازوں
بازی
بازیابی
باسط
باسی
باطل
باطن
باعث
باغ
باغات
باغوں
باغی
باقر
باقی
باک
باکس
بال
بالا تر
بالانی
بالائے
بالاصرار
بالر
بالرز
بالروں
بالز
بالغ

باولر
باونس
باونڈری
باب
بابا
بابائے
بابت
بابر
بابل
بابو
بابوں
باپ
باپردہ
باپوں
بات
باتونی
باتوں
باتیں
باجا
باجرے
باجوڑ
باجوں
باجہ
باجی
باجے
باد
بادام
بادامی
بادبان
بادباں
بادشاہ
بادشاہت
بادشاہوں
بادشاہی
بادل
بار
بارات
باراتیں
بارانی
باراں

اہانت
اہداف
اہل
اہم
آہٹ
آہیں
آیا
آیات
ایاز
ایام
آیت
ایدھی
ایذا
ایذائیں
ایران
ایرانی
ایرپورٹ
ایریا
ایریاز
ایریل
ایک
ایوارڈ
ایوان
ایوب
آیوڈین
ایویں
ایڈرس
ایڈز
ایڈمرل
ایڈوانزر
ایڈوانس
ایڑھی
ایڑی
ایڑیاں
اے
آئے
ب
با
با وفا
با وقار

پوریں
پورے
پوسٹ
پوش
پوشاک
پوشی
پول
پولیو
پونی
پونے
پوٹھوہار
پوڈر
پوڑیاں
پوڑیوں
پی
پٹ
پٹے
پڑنے
پڑا
پڑاو
پڑتا
پڑتال
پڑتی
پڑتیں
پڑتے
پڑنا
پڑوس
پڑوسن
پڑنے
پڑھا
پڑھائی
پڑھو
پڑھوا
پڑھوانے
پڑھوانا
پڑھوں
پڑھی
پڑھائیں
پڑھاتا
پڑھاتے

پروان
پروانہ
پرواہ
پرودر
گار
پرورش
پروقار
پروگرام
پرونا
پروٹین
پروں
پرویا
پروین
پروئے
پردہ
پردے
پرریونق
پرزور
پرزوں
پرزہ
پرس
پرست
پرسوز
پرسوں
پرکار
پرکالہ
پرلے
پرنالہ
پرند
پرندہ
پرندے
پرنور
پروئی
پروئیں
پروا
پرواز
پروازو
ں
پروازیں
ں

پرامن
پرانا
پرانده
پراٹھے
پرایا
پریت
پرتھوی
پرجوش
پرچا
پرچم
پرچون
پرچہ
پرخار
پرخاش
پرکی
پالا
پالتا
پالتو
پالک
پالنا
پامال
پان
پانچ
پاندان
پانڈا
پانی
پایا
پائے
پدر
پذیر
پذیرائی
پر
پرانز
پرانزبانڈ
پرائی
پرائے
پراپرٹی
پرات
پراثر
پراسرار
پراسراریت
پراگراموں

پارسل
پارک
پارلر
پاروں
پارٹی
پارہ
پازیب
پاس
پاسدار
پاش
پاک
پاکبازی
پاکدامن
پاکی
پالا
پالتا
پالتو
پالک
پالنا
پامال
پان
پانچ
پاندان
پانڈا
پانی
پایا
پائے
پدر
پذیر
پذیرائی
پر
پرانز
پرانزبانڈ
پرائی
پرائے
پراپرٹی
پرات
پراثر
پراسرار
پراسراریت
پراگراموں

بھونڈے
بھونیں
بھونے
بھی
بے دلی
بے دم
بے سود
بے ہوش
بے تاب
بے جا
بے خوف
بے زار
بے شک
بے قرار
بے کاری
بے گناہ
بے موت
بے موقع
بے نام
بے نور
بے وفا
بے وقت
بے وقوف
پ
پا
پاو
پانپ
پانداری
پاندان
پاونڈ
پاوڈر
پائی
پائیں
پائے
پاپڑ
پاتا
پاجامہ
پاداش
پادری
پار
پارسا

بڑھایا
بڑی
بڑے
بھائی
بھابھی
بھاپ
بھاتا
بھارت
بھاری
بھاشا
بھاگ
بھاگا
بھاگنا
بھاگو
بھاگی
بھالا
بھانپ
بھانت
بھانڈا
بھاو
بھاٹی
بھاڑ
بھوپال
بھوت
بھوتوں
بھورا
بھوسی
بھوک
بھوکا
بھول
بھایا
بھولا
بھولتا
بھولنا
بھولی
بھولے
بھون
بھونک
بھونڈا
بھونڈی

بوتا
بوتل
بودا
بودہ
بور
بوری
بوریاں
بوریت
بول
بولا
بولنا
بولو
بولوں
بولیاں
بولیوں
بولیں
بونا
بوند
بوندیں
بوٹا
بوٹی
بوڑھا
بوڑھوں
بوڑھی
بوڑھے
بویا
بوئے
بڈھا
بڑ
بڑا
بڑانی
بڑبڑا
بڑبڑایا
بڑوں
بڑھانیں
بڑھائے
بڑھاپا
بڑھاپے
بڑھاتا
بڑھانا

جانے
جابر
جابرانہ
جابوں
جاپان
جاتا
جاتیں
جاتے
جادو
جادوئی
جادوگر
جادوگری
جارحانہ
جاری
جاسوس
جاسوسوں
جاگ
جاگا
جاگتا
جاگزیں
جاگنا
جاگی
جاگے
جال
جالا
جالوں
جالی
جالے
جام
جامع
جامن
جانب
جانچ
جاندار
جانداروں
جانو
جانور
جانوروں
جانی
جانیں
جانے

تھال
تھالی
تھالیاں
تھالیوں
تھام
تھاما
تھان
تھانوں
تھانہ
تھانے
تھوک
تھوکنا
تھوڑا
تھوڑی
تھوڑے
تھی
تھے
تہ
تی
ٹ
ٹا
ٹابت
ٹاقب
ٹالث
ٹانوی
ٹانی
ٹروت
ٹریا
ٹواب
ٹور
ٹے
ج
جا
جاو
جانداد
جانداریں
جانز
جانزہ
جائی
جائیں

تواریخ
توازن
تواضع
توانا
توانائی
توبہ
توپ
توپیں
تودوں
تودے
توسط
توسل
توقع
توقف
توکل
تول
تولنا
تولو
تولہ
توند
توڑ
توڑا
توڑپھوڑ
توڑتا
توڑتے
توڑنا
توڑو
توہم
توہین
توے
تویج
ترین
تڑپ
تڑپانا
تڑپیں
تڑکا
تڑوا
تڑوانے
تڑی
تڑیاں
تھا
توا
تواتر

ترانے
تراویح
تربت
تربوز
ترپال
تردد
تردید
ترس
ترسا
ترسائے
ترسانا
ترسوں
ترسی
ترسے
ترش
ترشوا
ترشی
ترقی
ترک
ترک
ترکاری
ترکاریاں
ترکش
ترکہ
ترکی
ترکے
ترمذی
ترنگ
ترنم
تروپج
تزئین
تف
تک
تل
تم
تن
تو
توا
توانر

تاکنا
تال
تالا
تالاب
تالو
تالہ
تالی
تالیاں
تالیوں
تالے
تاش
تامل
تان
تانا
تاوان
تاویل
تاڑ
تاڑنا
تابم
تایا
تب
تدارک
تدبر
تدریج
تدریس
تدوین
تذبذب
تذکروں
تذکرہ
تذکرے
تر
تروتازہ
ترا
تراجم
ترازو
تراش
تراشا
تراشوانا
ترانوں
ترانہ

پی
ت
تا
تائی
تائے
تاب
تاباں
تابش
تابع
تابوت
تابوتوں
تاتاری
تاتاریوں
تاثر
تاثرات
تاثراتی
تاج
تاجدار
تاجر
تاجران
تاجروں
تاحال
تادیر
تار
تارا
تاراج
تارک
تارکول
تاروپود
تاروں
تارہ
تاری
تاریخ
تاریک
تارے
تازگی
تازہ
تازیانہ
تازے
تاک

پڑھاکو
پڑھانے
پڑھو
پڑھوا
پڑھوایا
پڑھوں
پڑھی
پڑھے
پڑی
پڑیا
پڑے
پھاند
پھاندنا
پھانس
پھانک
پھاٹک
پھاڑ
پھاڑا
پھاڑتے
پھوپھا
پھوپھو
پھوپھی
پھوس
پھول
پھولا
پھولنا
پھولوں
پھولی
پھولے
پھونس
پھونک
پھوٹ
پھوڑا
پھوڑتا
پھوڑنا
پھوڑنے
پھوڑو
پھوڑوں
پھوڑے
پھوہڑ

دربدر
درپردہ
درج
درجات
درجن
درجہ
درجے
درخت
درخواست
درد
دردانہ
دردناک
درزی
درزیوں
درس
درست
درسی
درشن
درکار
درکنار
درگاہ
درگت
درگذر
درندگی
درندہ
درندے
دروازوں
دروازہ
دروازے
درود
درویش
درہ
درہم
دری
دریا
دریاوں
دریائی
دریائے
دریافت
دریغ

داماد
دامن
داموں
دان
دانا
دانائی
دانت
دانش
دانہ
دانے
دائیں
دایاں
دب
دبا
دباو
دبائیں
دبانا
دبانے
دبایا
دبدبہ
داڑھی
دبک
دبوچ
دبوچا
دجال
دخل
در
دراز
درازوں
درازیں
دراصل
درآمد
درآمدات
دراڑ
دراڑیں
دربار
درباروں
درباریوں
دربان
دربانوں

خوشی
خوف
خوفزدہ
خوگر
خول
خونریز
خونریزی
خونیں
خوں
خون
د
دنے
دائر
دانروں
دائرہ
دائرے
دائم
دائی
دائیں
داب
دابیں
داتا
داخل
دادا
دادی
دادے
دار
داروغہ
داروں
داری
داریاں
داغ
داعی
داغدار
داغوں
داغے
دال
دالان
دام

خلوص
خم
خواب
خوابوں
خواتین
خواج
خواجہ
خوار
خواری
خواص
خواندگی
خواندہ
خواہ
خواہاں
خواہش
خوب
خوبرو
خوبی
خود
خودی
خودآگاہی
خودبین
خودبیں
خوددار
خودرو
خودسری
خودغرض
خودفراموشی
خودکش
خودی
خور
خوراک
خورد
خوردبین
خوش
خوش آمدید
خوشاب
خوشامد
خوشامدید
خوشامدیوں
خوردنی
خوشوں
خوشہ

خاوندوں
خداحافظ
خداداد
خدارا
خدوندی
خدوخال
خراب
خرافات
خرانٹ
خراٹے
خربوزہ
خرچ
خرچا
خرابی
خراش
خدارا
خدام
خداوند
خدایا
خدشات
خدشوں
خدشہ
خدمات
خدمت
خرگو
ش
خرم
خروج
خرید
خریدار
خریدنا
خریدنے
خرید و فروخت
خریدیں
خزائن
خزانوں
خزانہ
خزانے
خزاں
خط
خلوت

خادم
خادمہ
خار
خارج
خارجہ
خاردار
خارزار
خارش
خاشاک
خاص
خاصا
خاصہ
خاصی
خاطر
خاطرخواہ
خاطرداری
خاطرمدارات
خاک
خاکوں
خاکہ
خاکی
خالاوں
خالائیں
خالد
خالص
خالق
خالو
خالہ
خالی
خام
خاموش
خامی
خان
خانم
خانوں
خانہ
خانے
خاور
خاوند
خدا
خدائی

حدوں
حدود
حدیث
حدیں
حرا
حرارت
حراست
حرب
حربوں
حرج
حرص
حرف
حرکات
حرکت
حرکتوں
حرم
حرمت
حروف
حریص
حریف
حزف
حق
حل
حوا
حوادث
حواری
حواریوں
حوالات
حوالدار
حوالوں
حوالہ
حوالے
حور
حوروں
حوریں
حوض
خ
خائف
خاتم
خاتون

ذہن
ذہین
ذی
ذیل
ر
روسا
رانج
رائی
رائے
رات
راتوں
راتیں
راج
راجا
راجن
راجو
راجہ
راحت
راز
رازدار
رازدانوں
رازداں
رازق
رازوں
راست
راشد
راشن
راضی
راکٹ
راگ
ران
رانا
رانی
راوی
راہ
راہی
راہیں
رب
ربا
رباب

دیواروں
دیواریں
دیوالی
دیوان
دیوانوں
دیوانہ
دیوانی
دیوانے
دیوتا
دیومالا
دیومالائی
دیوی
دیویاں
دیں
دیے
دے
دئے
ذ
ذات
ذاتی
ذاتیں
ذاکر
ذبح
ذخائر
ذرا
ذرائع
ذرات
ذروں
ذرہ
ذرہ نوازی
ذری
ذرے
ذکر
ذلت
ذمہ
ذمہ دار
ذمے دار
ذوق
ذہانت
ذہد

دھونی
دھوئیں
دھویا
دہائی
دہانہ
دہانے
دہاڑا
دہاڑی
دہاڑیں
دہاڑے
دہرا
دہرائی
دہرائیں
دہرائے
دہراتا
دہراتی
دہراتے
دہرائیں
دہری
دہک
دہل
دہم
دہن
دہی
دی
دیا
دیار
دیانت
دیپ
دید
دیدار
دیدنی
دیدہ
دیر
دیرپا
دیس
دیگ
دین
دیو
دیوار

دوہا
دوہرا
دوہراتا
دوہرایا
دوہری
دوہرے
دھات
دھار
دھارا
دھاری
دھاریوں
دھاریں
دھارے
دھاک
دھاگا
دھاگوں
دھاگہ
دھاگے
دھان
دھاندلی
دھانی
دھاوا
دھاڑ
دھاڑیں
دھو
دھوئیں
دھوئے
دھواں
دھوبن
دھوبی
دھوپ
دھوتا
دھوتی
دھوتے
دھوکا
دھوکہ
دھوکے
دھول
دھوم
دھونا

دوران
دوراں
دوربین
دوررس
دوروں
دورہ
دوری
دوریاں
دوریوں
دورے
دوزخ
دوزخی
دوست
دوسرا
دوسروں
دوسرے
دوکان
دوکاندار
دوکانوں
دوکانیں
دوگانہ
دوگنا
دولت
دونوں
دوڑ
دوڑا
دوڑائی
دوڑائیں
دوڑائے
دوڑاتا
دوڑاتے
دوڑانا
دوڑتا
دوڑتی
دوڑتے
دوڑدھوپ
دوڑو
دوڑی
دوڑیں
دوڑے
دوں

دلاسا
دلاتا
دلاتے
دلاور
دلدادہ
دلدار
دلدل
دلربا
دلوا
دلوائی
دلوائیں
دلواتے
دلوانا
دم
دماغ
دماغوں
دمدار
دمک
دمڑی
دمہ
دن
دندان
دندانوں
دندناتا
دندناتی
دندناتے
دنوں
دو
دونم
دوا
دوائی
دوائیں
دوات
دواساز
دوبارہ
دوبالا
دوجے
دوچار
دوحہ
دور

دس
دشت
دعا
دعاوں
دعائیں
دعاگو
دعوت
دعوتوں
دعوتیں
دعوہ
دعویدار
دعوے
دغا
دغابازی
دف
دفاتر
دفاع
دفاعی
دفن
دق
دقت
دکان
دکاندار
دکاندارانہ
دکانوں
دکانیں
دکاں
دکن
دگنا
دل
دلاو
دلائل
دلاوں
دلائی
دلائیں
دلائے
دلاتا
دلاتی
دلاتے
دلارے

ساز
سازش
سازشوں
سازشی
سازگار
سازگاری
سازوسامان
ساس
ساعت
ساکت
ساگ
سال
سالا
سالار
سالانہ
سالم
سالن
سالی
سالیوں
سالے
سامان
سانپ
سانس
سانولا
سانولے
ساون
سایہ
سائے
سب
ساڑھی
سچ
سچ
سدا
سدباب
سر
سرا
سراپا
سراہا
سرایت
سربالا
سربراہ

زندگیاں
زندگیوں
زندہ
زوال
زوجہ
زور
زورآور
زوردار
زوروشور
زوری
زویا
زہر
زیادتی
زیادہ
زیارت
زیارتوں
زیارت
زیرہ
زیور
زیورات
ژ
س
سائرن
سائرہ
سائل
سات
ساتواں
ساتوں
ساتویں
ساتھی
ساجد
ساحر
ساحل
ساخت
سادگی
سادہ
سارا
سارہ
ساری
سارے

زاتی
زادراہ
زاویہ
زاویے
زاہد
زبان
زبانوں
زبانی
زبانیں
زباں
زبردست
زبور
زخم
زد
زدگان
زدوکوب
زرا
زراعت
زرافہ
زرد
زردہ
زردی
زرعی
زرہ
زری
زریں
زکات
زکاتیں
زکام
زکر
زلزلوں
زلزلہ
زلزلے
زلف
زمانوں
زمانہ
زمزم
زمین
زمیں
زندگی

رومال
رونا
رونق
روٹھا
روٹی
روڈ
رویا
رویہ
رٹا
رٹایا
رھا
رھانی
رھو
رھوں
رھی
رھے
رہا
رہائش
رہن
رہٹ
رہیں
رہے
ریاست
ریاضی
ریاکار
ریال
ریت
ریزہ
ریزے
ریس
ریل
ریوڑ
ریوڑیاں
ریٹ
ریڈیو
ریڑھی
ز
زائد
زائرین
زائل

رواج
روادار
روانہ
روانی
رواں
روایات
روایت
روپ
روپوش
روپے
روتا
روتی
روتے
روح
روحانی
روداد
روز
روزانہ
روزبروز
روزروشن
روزگار
روزمرہ
روزنامہ
روزوشب
روزوں
روزہ
روزی
روزے
روس
روسٹ
روسی
روشن
روضوں
روضہ
روضے
روغن
روک
روکا
ٹوکا
روکتا
روگ

رقت
رقص
رقم
رقوم
رک
رکا
رکاوٹ
رکاوٹوں
رکاوٹیں
رکتا
رکن
رکنا
رکو
رکوانی
رکوع
رکی
رکے
رگ
رگڑ
رگڑا
رگڑتے
رگڑنا
رلا
رلائے
رلاتا
رلاتا
رلایا
رمل
رموز
رموزاوقاف
رمی
رنج
رنڈوا
رو
روئی
رنگ
روئیں
روئے
روا
روابط

رباعی
ربانی
ربط
ربڑ
رت
رجب
رجوع
رحم
رخ
رد
ردا
ردوبدل
ردی
رزاق
رس
رسائل
رسائی
رسالت
رسالوں
رسالہ
رسالے
رسم
رسوا
رسوانی
رسوخ
رسول
رسولی
رسوم
رسومات
رسی
رش
رشک
رشوت
رضا
رفاعی
رفت
رفاقت
رفو
رقابت
رقاصہ

ضو
ضوابط
ط
طائر
طارق
طاعون
طاقت
طاقوں
طالب
طالبات
طالبان
طاہر
طاہرہ
طب
طرح
طرز
طرف
طرفدار
طواف
طور
طوطا
طوطوں
طوطی
طوطے
طوفان
طوفانی
طول
طویل
ظ
ظالم
ظاہر
ظاہری
ظرف
ظروف
ع
عابد
عاجز
عاجزانہ
عاجزی
عادات

صدف
صدقات
صدقہ
صدقے
صدموں
صدمہ
صدمے
صدی
صدیاں
صدیق
صدیوں
صراحی
صرافہ
صرف
صریح
صف
صوابدید
صوبائی
صوبوں
صوبہ
صوبے
صوتی
صور
صورت
صورتوں
صورتیں
صوفہ
صوفی
صوفے
ض
ضائع
ضد
ضدی
ضر
ضرب
ضرر
ضرور
ضرورت
ضرورتیں
ضروری
ضروریات

شاہد
شاہراوں
شاہراہ
شاہوں
شاہی
شاہین
شاید
شب
شد
شدت
شدید
شر
شرائط
شرابور
شرابی
شرارت
شرارتوں
شرارتی
شرارتیں
شرافت
شراکت
شربت
ص
صابر
صابرہ
صابن
صادر
صادق
صاف
صافی
صالح
صاحب
صد
صدا
صدائیں
صدارت
صدارتی
صداقت
صدام
صدر

سونے
سی
سڑ
سڑا
سڑک
سڑکوں
سڑنا
سڑوں
سہ
سے
ش
شاباش
شاباشی
شاخ
شاخوں
شاد
شاداب
شادابی
شادمانی
شادی
شادیاں
شادیوں
شاذونادر
شاذیہ
شارجہ
شارع
شاعر
شاعرانہ
شاعروں
شاعرہ
شاعری
شافی
شاگرد
شال
شام
شامت
شامل
شان
شاندار
شاہ

سوتے
سوجن
سوجی
سوچ
سوچا
سوچو
سود
سودا
سودائی
سوداباز
سوداگر
سودے
سور
سوراخ
سوراخوں
سورتوں
سورتیں
سورج
سورما
سوز
سوزش
سوغات
سوغاتوں
سوغاتیں
سوکن
سوگ
سوگوار
سوموار
سونا
سونامی
سونپ
سونف
سونی
سونے
سوٹ
سوڈا
سویا
سویاں
سویرا
سویر
ے

سرگرمی
سرگم
سرگودھا
سرگوشی
سرما
سرمائے
سرمایہ
سرمہ
سرو
سرور
سرورق
سروس
سرہانا
سرہانہ
سرہانے
سریا
سرے
سزا
سزائیں
سزاوار
سل
سن
سو
سوئم
سوئی
سوئیں
سوئے
سوا
سوائے
سوات
سوار
سواری
سواریاں
سواریوں
سوال
سوالات
سوالنامہ
سوالوں
سوانح
سوانے
سوتا

سربراہان
سربراہی
سرپرست
سرجری
سرجن
سرحد
سرحدو
ں
سرحدی
سرحدیں
سرخ
سرخاب
سرخی
سرد
سردار
سرداری
سردرد
سردی
سردیاں
سرزد
سرزمین
سرزنش
ش
سرسری
سرسوں
سرشار
سرطان
سرعام
سرفراز
سرقہ
سرک
سرکاتے
سرکار
سرکاردوعالم
سرکاری
سرکردہ
سرکس
سرکش
سرکوبی
سرکہ
سرگرداں
سرگرم

قاتل
قادر
قارئین
قارون
قاری
قاسم
قاصد
قاصر
قاضی
قاعدہ
قاعدے
قانون
قد
قدر
قدرت
قدردان
قدروں
قدریں
قدرے
قدسی
قدم
قرار
قدوقامت
قدیر
قدیم
قرارداد
قراقرم
قران
قرار
قرآنی
قرآں
قرب
قربان
قربانی
قرض
قرضوں
قرضہ
قرضے
قرعہ
قریب

فرمودات
فرمودہ
فروخت
فروری
فروغ
فرہاد
فریاد
فریادی
فریال
فریب
فرید
فریش
فریق
فزکس
فق
فن
فوائد
فوارہ
فوت
فوج
فورس
فورسز
فورم
فوری
فوجی
فولاد
فوم
فون
ق
قا
قائد
قائدانہ
قائدگی
قائدہ
قائدین
قائل
قائم
قابض
قابل
قابو

فراوانی
فراٹے
فراڈ
فراہم
فرج
فرح
فرحان
فرحت
فرخ
فرد
فردوس
فردوسی
فردیں
فرزانہ
فرزند
فرزندی
فرست
فرسودہ
فرش
فرشی
فرصت
فرض
فرضی
فرعون
فرق
فرقہ
فرقے
فرلانگ
فرم
فرمائش
فرمائی
فرمائے
فرماتا
فرماتی
فرمان
فرمانا
فرماتروا
فرماں
فرمایا
فرمائے

فائدہ
فائدے
فائرنگ
فائز
فائزہ
فاتح
فاخرہ
فارغ
فاروق
فاروقی
فاریہ
فاسق
فاش
فاضل
فاعل
فاقوں
فاقہ
فاقی
فاقے
فال
فالنامہ
فالتو
فانوس
فالج
فانی
فدا
فدائی
فرائض
فرانی
فرات
فراخ
فراخی
فرار
فراز
فراست
فراش
فراغت
فراق
فراموش
فراموشی

غالب
غدار
غداروں
غداری
غدر
غذا
غذاوں
غذائی
غذائیں
غراتے
غرارہ
غرارے
غرانے
غربا
غربت
غرض
غرق
غرناطہ
غروب
غرور
غریب
غزالہ
غزالی
غزل
غزلوں
غزلیات
غزوات
غزوہ
غزہ
غش
غل
غم
غور
غوری
غوطہ
غوغا
غٹ
ف
فا
فائدوں

عرش
عرصہ
عرصے
عرض
عرضی
عرف
عرفات
عرفان
عرق
عروج
عزائم
عزت
عزتوں
عزتی
عزم
عزیز
عزیزآباد
عزیزان
عزیزم
عزیزوں
عزیزہ
عش
عوام
عوامل
عوامی
عورت
عورتوں
عورتیں
عوض
غ
غا
غائب
غار
غارت
غارحرا
غاروں
غازی
غازیوں
غاصب
غافل

عادت
عادتوں
عادتیں
عادل
عادی
عار
عارضہ
عاری
عاشور
عاص
عاصم
عاطف
عاق
عاقل
عالم
عالی
عام
عامر
عدالت
عدالتوں
عداوت
عداوتوں
عداوتیں
عدت
عدد
عدل
عدم
عدنان
عدولی
عدی
عدیل
عذاب
عذابوں
عذر
عراق
عراقی
عرب
عربوں
عربی
عرس

گاڑتے	کڑا	کوائف	کرتیں	کاموں	کارندہ	قزاح
گاڑھا	کڑاکے	کوارٹر	کرتے	کان	کارندے	قل
گاڑھی	کڑاہی	کوتاہی	کردار	کانپ	کاروائی	قواند
گاڑھے	کڑتا	کوثر	کرسی	کاندھا	کاروان	قواعد
گاڑی	کڑوا	کوچ	کرکٹ	کاندھوں	کارواں	قوالوں
گاڑیاں	کڑواہٹ	کوچوان	کرم	کانوں	کاروبار	قوالی
گاڑیوں	کڑوی	کوچوں	کرن	کاوش	کاروں	قوالیاں
گپ	کڑوے	کوچہ	کرنا	کاٹا	کارٹون	قوالیوں
گپوں	کڑھانی	کود	کرنل	کاٹن	کارڈ	قوانین
گت	کڑی	کودا	کرنوں	کاٹی	کاریں	قوانیں
گداگر	کڑیاں	کودنا	کرنٹ	کاٹیں	کاش	قوت
گدگدی	کہ	کودنے	کرنیں	کابل	کاشت	قورمہ
گدھا	کی	کورا	کرنے	کب	کاشف	قورمے
گدی	کیا	کورس	کرو	کپ	کاشی	قول
گدے	کیوں	کوشاں	کروا	کدو	کاظم	قوم
گر	کے	کوشش	کرنی	کدورت	کاغذ	قوموں
گراونڈ	کی	کوکب	کرواتا	کدورتیں	کاغذات	قومی
گرانی	گ	کورے	کرواتی	کر	کاغذوں	قومیں
گرام	گا	کومل	کرواتیں	کرانی	کاغذی	قوی
گرامر	گاون	کونا	کرواتے	کرائیں	کاف	قے
گرانا	گاوں	کونے	کروانا	کرائے	کافر	ک
گرانی	گانی	کوٹ	کروانی	کراتا	کافروں	کا
گرانے	گانے	کوٹھا	کروانے	کراتی	کافوری	کابل
گراونڈ	گاتا	کوٹھوں	کروایا	کراتیں	کافی	کاپی
گراں	گاتی	کوٹھی	کروٹ	کراتے	کاکا	کاتب
گرایا	گاتیں	کوٹھے	کروٹیں	کراچی	کاکی	کاج
گرتا	گاجر	کوڑی	کروڑ	کرامات	کال	کاجل
گرتی	گاجروں	کوڑے	کروں	کرامت	کالا	کاجو
گرتے	گاجریں	کوہ	کریم	کرانا	کالاباغ	کار
گرج	گارا	کوہاٹ	کریں	کراٹے	کالج	کارتو
گرجا	گارے	کویت	کزن	کراہت	کالک	س
گرد	گال	کوے	کس	کرایا	کالم	کارروانی
گرداب	گالوں	کٹ	کل	کرایوں	کالو	کارواں
گردان	گامزن	کٹا	کم	کرایہ	کالونی	کارساز
گردانیں	گانا	کٹائی	کن	کرائے	کالوں	کارفرما
گردش	گانوں	کٹائے	کوئل	کرب	کالی	کارکن
گردن	گانے	کٹواتا	کوئلوں	کرتا	کالے	کارکنوں
گردے	گاڈ	کٹواتے	کوئلہ	کرتب	کام	کارگل
گرفت	گاڑ	کٹوتی	کونی	کرتوتوں	کامران	کارناموں
گرگٹوں	گاڑا	کٹورا	کوا	کرتی	کامل	کارنامہ

مدارس
مداری
مداوا
مدت
مدتوں
مدتیں
مدثر
مدح
مدحت
مدد
مددگار
مدراس
مدغم
مدفن
مدفون
مدرس
مدرسوں
مدرسہ
مدعا
مدنی
مدہوش
مدیر
مدیروں
مذاق
مذاکرات
مذاہب
مذکر
مذکورہ
مذمت
مذہب
مذید
مر
مرا
مراحل
مراد
مراسم
مراعات
مراکز
مراکش
مرتا

ماشہ
ماضی
مال
مالک
مالدار
مالش
مالوں
مالٹا
مالی
مالیاتی
ماما
مامور
مامون
ماموں
مامی
مان
مانا
ماندہ
مانگ
مانو
مانوس
مانوں
مانی
مانیں
ماورائی
ماورائے
ماں
ماہ
ماہانہ
ماہر
ماہرانہ
ماہوار
مایوس
مایوسی
مایہ
مت
مد
مداح
مداحوں
مدارات

لڑنا
لی
لیا
لے
م
ما
مائل
مائیں
مابدولت
مابین
ماپ
مات
ماتم
ماتھا
ماجد
ماجرا
ماچس
ماحول
ماحولیات
ماخوذ
مادہ
مادی
مادیات
مادیت
مادے
مار
مارا
مارتے
مارچ
مار دھاڑ
مارشل
مارکہ
مارنا
مارنے
ماریں
ماریہ
مارے
ماسوائے
ماسی
ماش

لرز
لرزتا
لش پش
لرزتے
لق
لق و دق
لرزش
لگ
لو
لواحق
لوازمات
لوٹ
لوح
لوری
لوکی
لوگ
لوگوں
لولاک
لومڑی
لونگ
لونڈی
لوٹ
لوٹا
لوہا
لوہار
لٹو
لڑ
لڑائی
لٹ
لڑاکا
لڑکا
لٹا
لٹانا
لٹاتی
لٹایا
لڑکوں
لڑکی
لڑکیاں
لڑکیوں
لڑکے

لانی
لانیں
لانے
لات
لاتا
لاتوں
لاتی
لاتیں
لاتے
لاثانی
لاج
لاجواب
لاچارگی
لادا
لادتا
لازم
لازمی
لازوال
لاش
لاغر
لافانی
لال
لالچ
لانی
لانے
لاوا
لاوارثی
لاٹھی
لاڈ
لاڈلا
لاڑکانہ
لاہور
لایا
لایں
لائے
لب
لباس
لتاڑنا
لذت
لذیذ

گوالوں
گواہ
گواہی
گوجرانوالہ
گود
گودام
گورا
گورنر
گوشت
گوشوارہ
گوشوں
گوشہ
گول
گولا
گولز
گولڑہ
گولی
گولیاں
گوند
گوندنا
گوندھا
گوٹا
گوہر
گویانی
گڈریا
گڈمڈ
گڑبڑ
گڑگڑاہٹ
گڑگڑایا
گڑھا
گڑیا
گی
گیا
گیارھویں
گیارہ
گے
ل
لا
لاو
لائق
لائن

گرم
گرمی
گرنے
گروہوں
گریز
گرویدہ
گرہ
گرہن
گربوں
گرہیں
گری
گریاں
گرے
گزار
گزارا
گزارتا
گزارش
گزرا
گزربسر
گزرتا
گزرگاہ
گزرنا
گزریں
گزارنا
گزارو
گزاروں
گزارہ
گل
گم
گن
گناہ
گناہوں
گنوا
گنواوں
گنوار
گنوانا
گوادر
گوارا
گوارہ
گوالا

وارداتوں	نزلہ	ناکس	نابالغ	مورچے	مری	مرتی
وارداتیں	نزلے	ناکوں	ناپ	مورخہ	مریخ	مرچ
وارنٹ	نل	ناکہ	ناپاک	مورنی	مرید	مرچوں
وافر	نم	ناگوار	ناتا	موروثی	مریض	مرحم
واقع	نو	ناگ	ناتواں	موری	مریل	مرحوم
واقف	نواب	نالائق	ناتے	موزے	مریم	مرحومہ
والا	نواز	نالاں	ناجائز	موسم	مزا	مرد
والد	نواسوں	نالوں	ناچ	موصوف	مزاج	مردان
والدہ	نواسہ	نالہ	ناچا	موصوفہ	مزاح	مردانہ
والدین	نواسی	نالیاں	ناچار	موصول	مزار	مردہ
والیاں	نوازشریف	نالے	ناچاقی	موضوع	مزاق	مردے
والیوں	نواح	نام	ناحق	موقع	مزدور	مرزا
والے	نواسے	نامزد	ناخن	موقف	مزکر	مرشد
واہ	نوافل	ناموافق	ناخوش	مول	مزہ	مرض
وبا	نوالا	نامور	نادار	مولا	مزہب	مرضی
وبال	نوبت	ناموس	نادان	مولانا	مزید	مرعوب
وتر	نوبل	ناموں	نادانوں	مولوی	مزیدار	مرغ
وثوق	نوجوان	نان	نادانی	مولی	مزے	مرغا
وجود	نودولتاتوں	نانا	ناداں	موم	مژدہ	مرغزار
وجوہ	نورانی	نانی	نادر	مومن	مل	مرغن
وجوہات	نوکدار	ناواقف	ناراض	مونس	من	مرغوب
وجہ	نوکر	ناول	نادرہ	مونگ	مواخذہ	مرغی
وحدت	نوکرانی	نایاب	ناریل	موٹا	مواد	مرغے
وحی	نوکروں	ندا	ناز	موٹاپا	موازنہ	مرکب
وراثت	نومین	ندامت	نازک	موٹر	موافق	مرکبات
ورثا	نڈر	ندی	ناسازگا	موٹرن	مواقع	مرکز
ورثہ	نڈھال	ندیم	ر	موڑ	موبائل	مرکزی
وردی	نہ	نذر	ناسور	موڑا	موت	مرکوز
وردیاں	نی	نذرانہ	ناصر	مٹ	موتی	مرگ
ورزش	نے	نذیر	ناطہ	مٹل	موج	مرگی
ورق	و	نر	ناطے	مڑا	موجزن	مرمت
ورقہ	وائرس	نرالا	ناظرہ	مڑوڑ	موجود	مرمر
ورم	واپس	نرالے	ناظرین	مڑوڑتا	موجودہ	مرنا
ورنہ	واپڈا	نرخ	ناغہ	میں	موچ	مروت
وزارت	واجب	نرسوں	ناقابل	ن	موچی	مروج
وزارتیں	واحد	نرم	ناقص	نا	مودبانہ	مروجہ
وزرا	وادی	نزاکت	ناک	ناو	موذی	مروڑ
وزن	وادیاں	نزاکتوں	ناکارہ	نائب	مور	مروڑا
وزنی	وارثین	نزدیک	ناکافی	نائی	مورت	مروں
وزیر	واردات	نزع	ناکام	نااہل	مورتی	مروی

ہڑپہ
ہڑتال
ہڑتالوں
ہیں
ہے
ی
یا
یاد
یادداشت
یاددہانی
یادگار
یادگاروں
یادوں
یادہانی
یادیں
یار
یاسر
یاقوت
یاور
یخ
یرقان
یزید
یورپ
یورپین
یوم
یونس
یہ
ے

ہدایت
ہدف
ہدیہ
ہر
ہرا
ہراساں
ہرجائی
ہرحال
ہرطرف
ہرگز
ہرن
ہرنی
ہری
ہریالی
ہزار
ہزاروں
ہزارہ
ہم
ہونی
ہوئے
ہوا
ہوائی
ہوائیں
ہوادار
ہوتا
ہوتی
ہوتے
ہولناک
ہونٹ
ہوٹل
ہی
ہڈی
ہڈیاں
ہڈیوں
ہڑپ

ڈھائی
ڈھائے
ڈھارس
ڈھاکہ
ڈھانپ
ڈھول
ڈھولک
ڈھونگ
ڈھونڈھ
ی
ڈیرہ
ڈیرے
ڈیل
ڈیم
ڑ
ھو
ھوتا
ھوں
ھوئے
ہاتھوں
ہاتھی
ہار
ہارتا
ہاروت
ہارون
ہاروں
ہاشم
ہاکی
ہال
ہانپ
ہانک
ہانڈی
ہاں
ہت
ہدایات
ڈ

ڈالیوں
ڈالے
ڈانٹ
ڈاڑھی
ڈرائیں
ڈراتی
ڈراما
ڈرامائی
ڈراموں
ڈرامے
ڈرانا
ڈرانے
ڈربے
ڈرپوک
ڈرتا
ڈرم
ڈرنے
ڈرو
ڈری
ڈس
ڈسا
ڈکار
ڈکارتا
ڈگری
ڈگریاں
ڈلوا
ڈلوانے
ڈلواتا
ڈلواتی
ڈلواتے
ڈنڈا
ڈنڈی
ڈنڈیاں
ڈنڈے
ڈٹ

ٹوکا
ٹوکتا
ٹوکرا
ٹوکروں
ٹوکری
ٹوکریوں
ٹوکرے
ٹوٹ
ٹوٹا
ٹوٹل
ٹوٹے
ٹڈی
ٹھاکر
ٹھوس
ٹھوک
ٹھوکر
ٹھوکریں
ٹھونس
ٹھونک
ٹھوڑی
ٹین
ڈ
ڈاک
ڈاکا
ڈاکو
ڈاکوں
ڈاکیا
ڈاکے
ڈالا
ڈالتا
ڈالر
ڈالنا
ڈالو
ڈالی
ڈالیاں

ولد
ولدیت
ولولہ
ولی
ووٹ
ووٹوں
وڈیرا
وڈیروں
وڈیرے
وہ
وہاب
وہابی
وہاڑی
وہاں
وہم
وہی
وہیں
ویران
ویرانوں
ویرانہ
ویزا
ویزہ
ویزے
ٹ
ٹاپوں
ٹارچ
ٹارزن
ٹارگٹ
ٹال
ٹالا
ٹالنا
ٹانگ
ٹب
ٹرین
ٹوپی

وزیرآباد
وزیرخارجہ
وزیرخزانہ
وزیروں
وسائل
وساطت
وسط
وسوسوں
وسوسہ
وسوسے
وصال
وصول
وضاحت
وضو
وطن
وظائف
وعدوں
وعدہ
وعظ
وفا
وفات
وفادار
وفاق
وفاقی
وفد
وفود
وقار
وقاص
وقت
وقف
وقوع
وقوعہ
وقوف
وکٹ
وکٹوریہ
ولادت

Total: 4198 words

APPENDIX 3

Introduction:

The research has been implemented in Visual C# dot net. Some of the important code segments are given here.

1 Smoothing:

```
private void smothing(ref int[] dir,ref int n)
{
   for( int i=0; i < n; i++)
    {
      //////////////1 pixel smoothing//////////////
      if(i==0 || i==n-1)
            continue;
       else if(dir[i-1]==dir[i])
       {
            if(dir[i+1]==dir[i])
                  continue;
            else if(dir[i+1]!=dir[i])
            {
                  if(dir[i+2]==dir[i])
                        dir[i+1]=dir[i+2];
                  else if(dir[i+1]!=dir[i+2])
                  {
                        if(dir[i+2]==dir[i+3])
                              dir[i+1]=dir[i];
                  }
                  else
                  {
                        if(dir[i+1]==dir[i+2])
                        {
                              if(dir[i+2]==dir[i+3])
                                    continue;
                              else if(dir[i]==dir[i+3])
                              {
                                    dir[i+1]=dir[i];
                                    dir[i+2]=dir[i];
                              }
                        }
                  }
            }
            else if(dir[i]!=dir[i+1])
            {
                  if(dir[i]==dir[i+2])
                        dir[i+1]=dir[i];
                  if(dir[i+1]!=dir[i+2])
                  {
                     if(dir[i+2]==dir[i+3])
                     {
                        dir[i]=dir[i-1];
                        dir[i+1]=dir[i-1];
                     }
```

```
                    else if(dir[i]==dir[i+3])
                    {
                        dir[i+1]=dir[i];
                        dir[i+2]=dir[i];
                    }
                }
                else if(dir[i+1]==dir[i+2])
                        dir[i]=dir[i-1];
            }
            else if(dir[i]==dir[i+1])
            {

                if(dir[i]==dir[i+2])
                        continue;
                else
                {
                        dir[i]=dir[i-1];
                        dir[i+1]=dir[i-1];
                }
            }
        }
    }
}
```

2 Dehooking:

```
private void dehooking(ref int[] dir,ref int n)
{
    int tempI=0;
    ////////////////Dehooking
    if(dir[0]!=dir[1])
    {
        if(dir[1]==dir[2])
            dir[0]=dir[1];
        else
        {
            dir[0]=dir[2];
            dir[1]=dir[2];
        }
    }
    else if(dir[0]==dir[1])
    {
        if(dir[1]==dir[2])
            dir[1]=dir[2];
        else if(dir[1]==dir[3])
            dir[2]=dir[3];

    }
    if(dir[n-1]!=dir[n-2])
    {
        if(dir[n-2]==dir[n-3])
            dir[n-1]=dir[n-2];
```

```
            else
            {
                  dir[n-1]=dir[n-3];
                  dir[n-2]=dir[n-3];
            }
            else if(dir[n-2]!=dir[n-3])
                  n=n-2;
      }
}
```

3 Chain Coding:

```
private void chancode(ref int[] XIndex, ref int[] YIndex,int cnt)
{
   if(lastdir==0)
         n1=1;
   for( int i=0; i < cnt; i++)
   {
         if(XIndex[i]==XIndex[i+1])
         {
               if(YIndex[i]<YIndex[i+1])
                     dir[n++]=6;
               else
                     dir[n++]=2;
         }
         else if(YIndex[i]==YIndex[i+1])
         {
               if(XIndex[i]<XIndex[i+1])
                     dir[n++]=0;
               else
                     dir[n++]=4;
         }
         else if(XIndex[i]<XIndex[i+1])
         {
               if(YIndex[i]<YIndex[i+1])
                     dir[n++]=7;
               else
                     dir[n++]=1;
         }
         else if(XIndex[i]>XIndex[i+1])
         {
               if(YIndex[i]<YIndex[i+1])
                     dir[n++]=5;
               else
                     dir[n++]=3;
         }
   }
```

4 Check Vertical:

```
private void ChkVer(ref int[] dir,ref int n)
{
   int Stcntr=0;
   for(int i=0;i<n;i++)
```

```
    {
          if(dir[i]==6||dir[i]==7||dir[i]==5)
                Stcntr++;
    }

    if(n-cntr<2)
          Stver=1;
 }
```

5 Check Horizontal:

```
private void ChkHor(ref int[] dir,ref int n)
{
    int L2Rcntr=0;
    int R2Lcntr=0;
    for(int i=0;i<n;i++)
    {
          if(dir[i]==0||dir[i]==7||dir[i]==1)
                L2Rcntr++;
          if(dir[i]==3||dir[i]==4||dir[i]==5)
                R2Lcntr++;
    }
    if(L2Rcntr>8)
          HorL2R=1;
    if(R2Lcntr>8)
          HorR2L=1;
 }
```

6 Loop Up:

```
 private int LoopUp()
 {
    int lUp=0;
    int cntY=0;
    int cntX=0;
    int difI=3;
    int j=0;
    cntY=YIec(ref i);
    cntX=XIec(ref j);
    if(endloop1<=4)
          difI=cntY;
    if(((difI-cntY)<=1)&&((difI-cntY)>-3))
    {
          if(cntX>=2)
                lUp=1;
          else
                lUp=0;
    }
    else
          lUp=0;

    if(cntY>=5)
    {
```

```
            lUp=0;
            corArray[index].stVer=1;
        }
        return lUp;
    }
```

7 Unicode:

```
private void Unicodes(ref string unicode,string arr)
{
      switch(arr)
      {
            case "Alif":
                  unicode="\u0627";
                  break;
            case "Tthay":
                  unicode="\u0679\u06BE";
                  break;
            case "Nhay":
                  unicode="\u06BA\u06BE";
                  break;
                  :
                  :
                  :
                  :
            default:
                  break;
      }
}
```

8 Word Formation:

```
private void WrdFormation()
{
      if(Lig[LigIndx-1].XCor[end]-Lig[LigIndx].XCor[0])<20
            LigIndx++;
      else
      {
      for(int i=0;i<LigIndx;i++)
            Wrd[wrdIndx].Lig[i]=Lig[i];
      Wrd[wrdIndx].LigIndx=LigIndx;
      wrdIndx++;
      }
}
```

9 Check Validity:

```
private void ChkValidity()
{
      for(int i=0;i<dicIndex;i++)
      {
```

```
            if(Dic.Unicode[i].CompareTo(curWrd.Unicode)==0)
            {
                  valid=1;
                  break;
            }
      }
      if(valid==1)
            SaveFile(curWrd.Unicode);
      else
      {
            Delete(curWrd.Unicode);
            MessageBox.Show(`Enter Valid Wrd Again`);
      }
}
```

Online Urdu Character Recognition System

S. A. Husain, Asma Sajjad, Fareeha Anwar
drafaq@iiu.edu.pk
Dept of Computer Science, Faculty of Applied Science, International Islamic University, Islamabad.

Abstract

The reduction in the prices of technology has increased the use of handheld devices. These devices also provide the facility of text input but these provide an inconvenient way of input i.e. the very small keypads. Digitizing tablets on the other hand provide a natural and convenient way of input. There are many online character recognizers for many languages but there is no such commercial product for Urdu. Our research deals with the online Urdu handwriting recognition.

Our online Urdu character recognizer makes use of the ligature based approach instead of character based identification. The segmentation free system extracts a feature vector for each ligature which is then passed on to the neural network for classification of the ligature. We have used the back propagation neural network. The special ligatures (Dots, Tay, Hamza, Diagonal & Mad) are identified from the base ligatures. These special ligatures are associated with the base ligature. After this, the ligature is checked for its validity. Valid ligatures form words. After word formation, word validity is checked by using a word dictionary. Finally, the valid words are written in a text file.

Keywords: Online Character Recognition, Urdu handwriting recognition, Ligature based identification, digitizing tablets, handwritten characters, feature extraction, Back-Propagation Neural Network.

1. INTRODUCTION

The **user interface** is the aggregate of means by which people interact with a particular machine, device, computer program or other complex tool (*the system*). More and more efforts are being made on both the software and hardware side in order to make this human computer interaction more and more friendly.The development of pen interfaces is a key element in providing an efficient and natural way of input to the computer. These pen devices have also increased the market for lightweight, handheld computers. For example, present PDAs usually have a graphical user interface in which a pen can be used for pointing and selection functions, drawing, and text entry.

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 touch pads, where the written data is digitized and translated into a series of coordinates.

Urdu is the national language of Pakistan. The hand held devices have also successfully emerged in Pakistan but the software they provide for user input are mostly in English. Where as the common man in Pakistan can not communicate in English easily. In order to reduce this difference between the common man and the new technology Urdu input software were required. Our research is a step in order to bridge this gap. Urdu handwriting recognition is a complex process due to the complex nature of the Urdu script. The complexities that this language poses compared to the other international languages are:-

Cursive: Urdu text is cursive in nature [10]. Adding to the complexity is the writing style in which the characters forming words are connected to each other.

Ligatures: Several characters of Urdu are combined vertically to form a ligature [10].

Spaces: Spaces in Urdu may occur between ligatures and between words. The spaces between ligatures and words vary. This feature of Urdu handwriting also makes the recognition difficult.

Overlapping: The recognition of individual characters with in a ligature becomes quite difficult as the characters in Urdu overlap vertically and do not touch each other.

Diacritics: Diacritics are very important in Urdu language. These include diacritics such as: *Dots, Tay, Hamza, Diagonal & Mad*, etc. [16].

Context sensitivity: Every character in Urdu can have up to 4 different shapes (in Nasakh Font) depending on its position with in a ligature i.e. whether the character is isolated, in the beginning, at the end or connected from both sides in a word [10].

Strokes: The basic rule is that any Urdu character has one main stroke and zero or one secondary stroke

Direction of writing: Unlike English, Urdu is written from right to left. [10].

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.

It is the complexity of Urdu script which poses great challenges to the new field of online Urdu handwriting recognition.

2. PREVIOUS WORK

There are many offline OCR systems available for handling printed Arabic and Urdu documents with reasonable levels of accuracy. However, there are not many reported efforts of developing online (dynamic) OCR systems for Urdu language. This may be due to the complexities involved in the online character recognition with the added difficulties of Urdu 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.

Samir Al-Emmy and Mike Usher [20] have recognized only four words (اجی، علیہ، دار، زناق). In order to identify the characters in the words, segmentation approach is used. Learning was achieved by storing the stroke specifications in tree nodes. The recognition process is achieved through tree matching. The results were 100 % only for proper words. Variations in the words did not give appreciating results. S Malik and S.A.Khan, [18] have recognized only individual characters and Urdu numerals in their research but ligatures have not been addressed. Using the individual characters, 200, two character words were recognized. For example, ?? ?Q etc. In many on-line character recognition papers, for other languages, neural networks were used for recognition but they have used tree based dictionary search for the classification of characters. The recognition rate for the isolated characters and numerals is 93% and 78% for two character words. Another research in this field wa a research project completed at NUCES[19]. They have recognized words using the segmentation based approach. Characters were classified into 60 classes. This large number is due to the context sensitive shape of Urdu characters. Spatial Temporal Neural Network was used for recognition.

Only these systems exist in this emerging field of Online Urdu Character Recognition. The proposed system is an addition to this list.

3. PROPOSED METHODOLOGY

The recognition engine makes use of the various approaches in order to recognize the strokes. This is due to the cursive nature of the Urdu handwriting. The recognition systems are generally divided in to two types. Segmentation based and segmentation free recognition systems.

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. The segmentation free system extracts a feature vector for each ligature which is then passed on to the back propagation neural network for classification of the ligature. 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.

The secondary strokes recognized are:

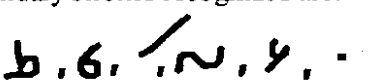


Figure 1: Secondary Strokes

Namely, (left-right order) small tuan, hay, kaaf and gaaf long diagonal stroke, Madaa, Hamzaa, and the single dot. The special ligatures are identified from the base ligatures. These special ligatures are associated with the base ligature. After this, the ligature is checked for its validity. Valid ligatures form words. After word formation, word validity is checked by using a word dictionary. Finally, the valid words are written in a text file. The OLUCR recognizes 38 one character ligatures , 709 two character ligatures e.g. ??, ?? etc and approximately 50 most commonly used three character ligatures e.g. ???, ???, ???, ???, ???etc.

The constraints for the system are that the base stroke should be written before the secondary stroke. For the two character ligature expecting a secondary stroke, the secondary stroke for the first

character should be written first and for the second character should be written second.

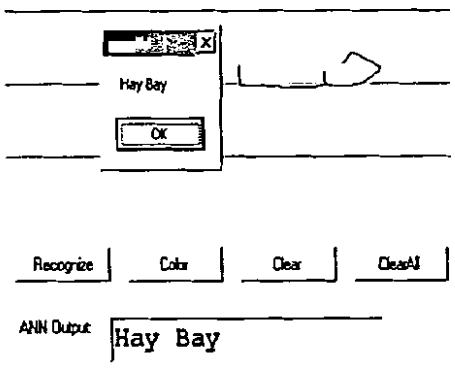


Figure 3a: Base Stroke

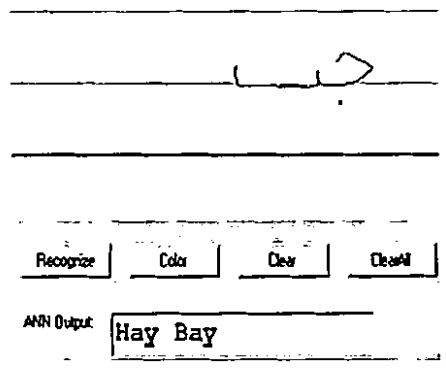


Figure 3b: secondary stroke for first character

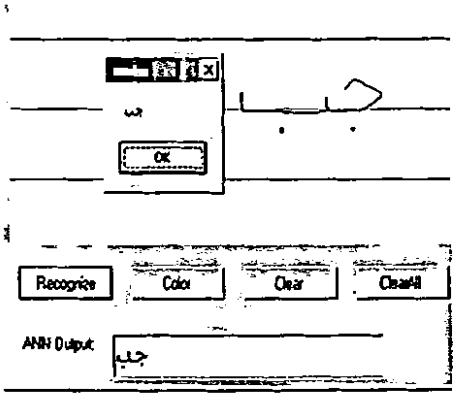


Figure 3c: secondary stroke for second character

The block diagram for the proposed model is given in figure 2.

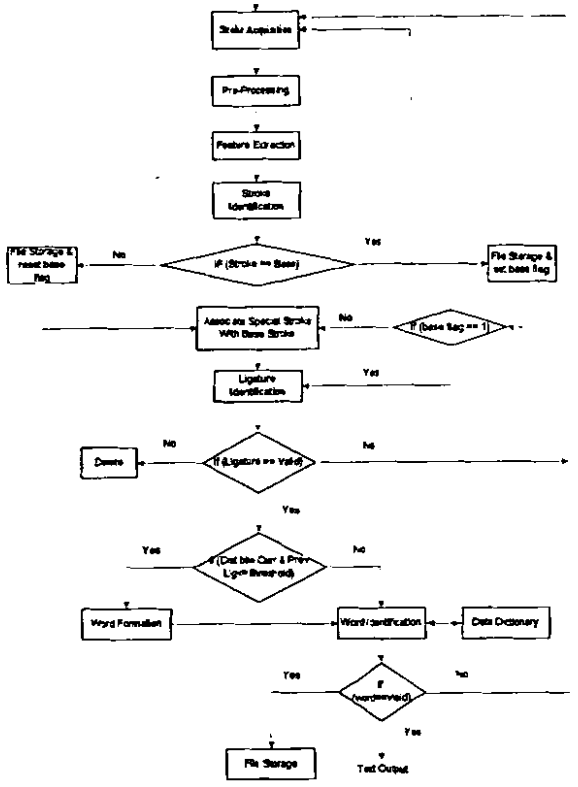


Figure 2: Block Diagram of Proposed System

3.1 Proposed 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 of many online character recognizers. The modules include:

3.2.1 Acquisition:

The processing was done on strokes obtained using digitizing tablet, the Intuos Wacom board. The data is collected in real time. The standard data is a stream of {x, y} coordinates. The data resolution is around 100 DPS (dots per second).

3.2.2 Preprocessing:

The data thus obtained often contains irregularity such as the hooks and erratic handwriting generated by inexperienced users. Hooks occur due to

the inaccuracies during pen up and pen down while placing the stylus on, or lifting it off the tablet.

Figure 4: ? written by inexperienced writer and ??containing hooks in the beginning and end.

3.2.3 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 features 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 ? feature which is selected on the presence of ? in any ligature. These features were very helpful in uniquely distinguishing the ligatures.

3.2.3.1 Features for Base Stroke:

The feature vector prepared for recognizing the base strokes includes the following features:-

Start Vertical: This feature was selected when the ligature was a straight vertical in the beginning e.g. ?, ?.

End Vertical: This feature was selected when the ligature was a straight vertical in the end e.g. ?

Horizontal R2L: If while writing the ligature the pen movement is from right to left horizontally then a bit is set for this feature vector e.g. in ?, ? .

Horizontal L2R: While writing the ligature, if the pen movement is from Left to right horizontally then this feature is set in the feature vector e.g. in ?, ? .

Hedge: In Urdu characters like, ?, a , ?, ? , ? a semi circle sort of shape is present in them which we call curve. For such characters we have selected a feature called the hedge.

Curve R2L: The direction of writing of these curves varies from right to left and also from left to right. Therefore, Curve R2L has been set for characters whose writing direction is right to left e.g. ?, ?.

Curve L2R: If the curve direction of the character from left to right then Curve L2R is set for them e.g. ? and ?

Loop Flag: Loops are very common features of Urdu handwriting. They are present in characters such as ?, ?, ?, ? and ? . Whenever the recognition engine finds a loop it selects this feature for that character or ligature.

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 a and ?? as shown in the figure below.

Figure 5: Cusp in character a

Intersection: When ever an intersection is encountered in a stroke this feature is selected for that stroke e.g. these are present in a , ?, ??etc.

Ray: This feature is selected for the character ray of Urdu alphabet. If any ligature is a combination of ray then this feature is also selected for that particular ligature e.g. ??, ??, ?? ??, ??,?? etc.

End Up Vertical: This feature is selected for the characters having a vertical end in the upward direction e.g. ?, ? , ? , ?, ?, ?, ? ? etc.

Loop Up: In order to differentiate the loop in as ?, ? and ?, this feature was identified and selected for ?. The writing direction of the loop in ? is from down to up as shown in figure below.

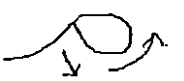


Figure 6: Writing direction of loop of ?

Seen Bit: This feature was selected if character seen is detected in any ligature e.g. ß ?, ? ?, ß ?, ß ? etc.

Aien Bit: This feature was selected if character ? is detected in any ligature e.g. ??,?? ,??, ,??,???? etc.

Hay Bit: The presence of hay in any ligature is shown by the selection of this bit in the feature vector e.g. ?S ,?? ,?? ,?? , ??, ?? etc.

Dal Bit: If the recognition engine detects a On the ligature written it selects this feature for it e.g. S, S ??,?? ?,?? ?, ?? s, s etc.

Double Loop: This feature is selected for characters and ligatures which have two loops in them e.g.

ligatures like ??, ??, ??, ? ?etc have this feature selected in their feature vector.

Figure 7: double loop ligatures

Tuan Bit: This feature is selected on the presence of ? in any ligature e.g. ??,?? ?,??, ?? ??? ?etc.
Gol Hay: All the gol hay ligatures have this feature selected for them e.g. ??,?? ,?? ,??, ?? etc. The shape of gol hay ligatures is given below

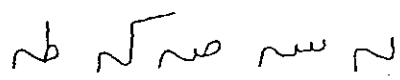


Figure 8: Gol Hay Ligatures

3.2.3.2 Features for Secondary Stroke:

The feature vector prepared for the secondary strokes includes the following features:-

Dot: If the secondary stroke has a length less then or equal to the threshold specified for a dot it is taken as a dot provided it is with in the boundaries of the base stroke. If a dot is encountered this feature is selected.
Madaa: If the stroke fulfills the criteria set for the madaa stroke. This feature is selected then.
Diagonal: This secondary stroke feature is selected for the diagonal stroke occurring in ? and _ . If any base ligature has a diagonal over it this feature is selected.

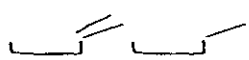


Figure 9: The diagonal stroke over ? and _

Hay: This feature is selected if the secondary stroke called the hay %s encountered. For e.g. in ?, ? ?

Figure 10: The hay stroke

Hamzaa: Hamzaa is a secondary stroke which is often present over the base strokes. If the secondary stroke is Hamzaa then this feature is selected.

Chooti Tuan: This secondary stroke is present over the base ligatures. If the loop follows a vertical line then the stroke is a ?. In this case this feature will be selected as a feature vector.

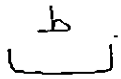


Figure 11: The ? over the base stroke kashti.

3.3 Stroke Identification:

This stage can be further sub divide in to two identification phases. The identification of the base stroke and that of the secondary stroke. This identification has been done using Back-Propagation neural network.

The network for the base ligature recognition consists of twenty inputs, sixty nine hidden nodes and One fifty three output neurons. The learning rate for this has been set to 0.6F. The network for the for this has been set to 0.5F. The number of hidden nodes and the learning rate were set through testing various values. Secondary strokes recognition consists of five inputs, five outputs and ten hidden nodes. The learning rate

3.4 Ligature Validation:

In this stage, the ligature written is check for its validity. If the ligature is valid then the next ligature is considered. Otherwise, the ligature is removed to be rewritten.

3.5 Ligature Combination/ Word Formation:

In this stage the difference between ligatures is considered. If the difference between the previous ligature and the current ligature is less then the threshold which is necessary for word formation then these ligatures form one word. When the difference between successive ligatures increases then the new ligature is taken up as another word. In this way a series of words can be written. . For example ?is one ligature and ?? is another ligature. When we combine these two, the word ??is formed.

3.6 Valid Word Identification:

This stage makes use of the dictionary to identify that the word written is valid or not. If the dictionary does not contain the word then the word is discarded. For example ?? is a valid word but ?? is

not a valid word. Therefore, the dictionary will not identify it and discard it.

The dictionary comparison is based on the words Unicode comparison. If the Unicode of the word written is present in the dictionary then the word is a valid otherwise it is considered an invalid word and removed from the interface. An invalid word is also not written in the file.

3.7 Output:

The output is the Urdu Text in the interface's text area and in a word file. Only the valid words 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. The writing is done by sending Unicode to the file through the program.

آپ کلم کرتے ہیں.

Figure 12: The text in file.

4. RESULTS

The system was trained using a training set of 240 carefully selected ligatures. With the combination of 6 diacritics we have successfully recognized more than 850 ligatures. These 850 ligatures form approx 18000 words. The Recognition rate of base ligatures was 93% and of the secondary strokes was 98%.

Test results of some of the difficult ligatures and diacritics (Aerab) are given in the tables below.

No	Ligature	Recognition Rate
1	٠	100 %
2	~	85 %
3	/	100 %
4	،	95 %
5	ط	95 %
6	؟	95 %

Table 1: Recognition Rate of diacritics w.r.t. samples

No	Ligature	Total Samples	Recognition Rate w.r.t. samples in %
1.	? class	18	89.3
2.	a class	18	90.6
3.	? class	18	87.3
4.	? class	18	93
5.	??	18	96.8
6.	?s	18	86.5
7.	?s	18	87.3
8.	??	18	87.3
9.	??	16	83.5
10.	?S	16	95.5
11.	??	16	95.5
12.	? ?	16	95.5
13.	??	16	95.5
14.	??	16	95.5
15.	??	16	85.7
16.	? ?	16	94.5
17.	?s	16	94.5
18.	??	16	91
19.	???	16	87.5
20.	??	16	96.5

Table 2: Recognition Rate of base ligatures w.r.t. samples

5. Concluding Remarks

In this paper, we have presented a method for recognition of online Cursive Urdu hand written Nastaliq Script. The system is currently trained for 250 ligatures. Our approach minimizes the errors due to segmentation by using segmentation free approach. By using multiple classes of features, we have improved the number of ligatures that can be identified. We have successfully recognized 250 base ligatures and 6 secondary strokes. These when combined form more than 850 ligatures which can recognize approx 18000 words of our Urdu dictionary successfully. The implementation was done in Visual C# dot net.

6. Future enhancements

As our research and implementation was an initial step. Therefore, there is a lot of scope for future enhancement.

- ? Implementation of other pre-processing techniques such as the RTS techniques.
- ? Enhancement in the number of ligatures which we think is a continuous area of

research. i.e. the recognition of 4 characters ligatures and so on.

? Recognition of additional secondary strokes such as the shad, zeer, zabar and paish.

? Recognition of Urdu numerals.

REFERENCES

1. http://www.ethnologue.com/show_language.asp?code=urd.

2. http://www.omniglot.com/writing/urdu.htm.

3. http://www2.psy.uq.edu.au/~brainwav/Manual/BackProp.html

4. http://std.dkuug.dk/JTC1/SC2/WG2/docs/n2413-3.pdf

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

6. M S Khorsheed, "Off-Line Arabic Character Recognition A Review".

7. Mohammad S. Khorsheed, "Automatic recognition of words in Arabic manuscripts", PhD Dissertation, Churchill College, University of Cambridge, June 2000.

8. H. Bunke, P. Wang, "Handbook of character recognition and document image analysis", World Scientific, 2000.

9. Syed Afaq Husain and Syed. Hassan Amin, "A Multi-tier Holistic approach for Urdu Nastaliq Recognition", INMIC 2002.

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

11. Sutat Sae-Tang Ithipan Methaste. "Thai Online Handwritten Character Recognition Using Windowing Backpropagation Neural Networks", Information Research and Development Division, National Electronics and Computer Technology Center, National Science and Technology Development Agency, Rachathewi, Bangkok 10400, Thailand.

12. VKazushi Ishigaki VHiroshi Tanaka VNaomi Iwayama. "Interactive Character Recognition technology for Pen-based Computers".

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

14. A.Amin, G. Masini and J.P. Haton, "Recognition of Handwritten Arabic Words and Sentences", 7th Int.Conf on Pattern Recognition, Montreal, 1984, 1055-1057.

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

16. I.Guyon, J.Bromley, N.Matic, etc, "A neural network system for recognizing on-line handwriting", Models of Neural network, Springer Verlag, 1996.

17. Sarmad Hussain and Muhammad Afzal, "Urdu Computing Standards", Urdu Zabta Takhti (UZT) 1.01 - WG2 N2413-3 / SC2 N3589-3

18. 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.

19. "Online Character Recognition using Artificial Spatial Temporal Neural Network", BS Project report, Nuces-Fast, Islamabad, 2003

20. 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