



INVESTIGATION INTO A SEGMENTATION BASED OCR FOR THE NASTALEEQ WRITING SYSTEM

MS Thesis

Submitted in Partial Fulfillment
Of the Requirements of the
Degree of

Master of Science (Computer Science)

**AT
NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES
LAHORE, PAKISTAN
DEPARTMENT OF COMPUTER SCIENCE**

**By
Sobia Tariq Javed
August 2007**

Approved:

Head
(Department of Computer Science)

Approved by Committee Members:

Advisor

Dr. Sarmad Hussain
Professor
FAST - National University

Other Members:

Dr. Mehreen Saeed
Assistant Professor
FAST - National University

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance of a thesis entitled "Investigation into a Segmentation Based OCR for the Nastaleeq Writing System" by Sobia Tariq Javed in partial fulfillment of the requirements for the degree of Master of Science.

Dated: August 2007

To my parents

Vita

Ms. Sobia Tariq Javed received a Bachelor of Science degree in Computer Science from National University of Computer and Emerging Science (NUCES), Lahore in 2005. Sobia Tariq Javed has been a member of CRULP since 2004. She has been working as a Research Officer in different R&D works. The research in this dissertation was carried out from 2006 to 2007.

Acknowledgements

First I would like to express my gratitude to Almighty Allah for the blessings He has bestowed on me.

I would like to thank Dr. Sarmad Hussain, Associate Professor at NU-FAST for his generous help, support and precious time that he gave me for the completion of this thesis. I would like to thank him for his insight and supervision.

I would also like to thank Dr. Mehreen Saeed for encouraging me to complete this research.

My special thanks to Ameera Maqbool for helping me in understanding HTK tool kit for Hidden Markov Model. She contributed a lot by providing all sorts of information regarding HTK that I needed.

And last and the most important people my parents who made me whatever I am today. Their prayers and continuous encouragement have led me to this point.

Sobia Tariq Javed

TABLE OF CONTENTS

| | | |
|-----------|--|----|
| 1 | OCR | 9 |
| 1.1 | The Need for OCR Systems..... | 9 |
| 1.2 | Generic Model of an OCR System | 9 |
| 1.2.1 | Image Acquisition..... | 10 |
| 1.2.2 | OCR Software..... | 10 |
| 1.2.2.1 | Document Analysis..... | 11 |
| 1.2.2.1.1 | Skew Detection and Removal..... | 11 |
| 1.2.2.1.2 | Binarization..... | 11 |
| 1.2.2.1.3 | Filtering and Smoothing | 12 |
| 1.2.2.1.4 | Thinning..... | 12 |
| 1.2.2.1.5 | Normalization | 12 |
| 1.2.2.2 | Character Recognition | 13 |
| 1.2.2.3 | Contextual Processing..... | 13 |
| 2 | Literature Review | 13 |
| 2.1 | Background and History | 13 |
| 2.2 | Research Techniques for Character Recognition..... | 14 |
| 2.2.1 | Segmentation | 14 |
| 2.2.1.1 | Segmentation Free Approach..... | 14 |
| 2.2.1.2 | Segmentation Approach..... | 15 |
| 2.2.1.2.1 | Segmenting into Characters | 16 |
| 2.2.1.2.2 | Segmenting into Primitives..... | 16 |
| 2.2.2 | Recognition..... | 18 |
| 2.2.2.1 | Feature Extraction..... | 18 |
| 2.2.2.1.1 | Structural Features | 19 |
| 2.2.2.1.2 | Statistical Features | 21 |
| 2.2.2.1.3 | Global Transformations | 21 |
| 2.2.2.2 | Pattern Matching / Classification..... | 22 |
| 2.2.2.2.1 | Template Matching..... | 22 |
| 2.2.2.2.2 | Statistical Classifier | 23 |
| 2.2.2.2.3 | Clustering..... | 28 |
| 2.2.2.2.4 | Nearest Neighbor | 28 |
| 2.2.2.2.5 | Syntactical / Structural..... | 29 |
| 2.2.2.2.6 | Expert System | 30 |
| 3 | Urdu Writing System | 31 |
| 3.1 | Character to Unicode and Glyph Mapping..... | 35 |
| 3.2 | Advance Script Properties | 36 |
| 3.2.1 | Character Composition and Decomposition | 36 |
| 3.2.1.1 | Different Shapes | 36 |
| 3.2.1.1.1 | Isolated Form | 36 |
| 3.2.1.1.2 | Initial Form | 36 |
| 3.2.1.1.3 | Final Form..... | 37 |
| 3.2.1.1.4 | Medial Form | 37 |
| 3.2.1.2 | Connection Form | 37 |
| 3.2.1.3 | Diacritic Placement..... | 38 |

| | | |
|---------|--------------------------------------|----|
| 3.2.1.4 | Presence of Base Line..... | 38 |
| 3.2.1.5 | Overlapping | 38 |
| 3.2.1.6 | Upper and Lower Cases..... | 39 |
| 3.2.1.7 | Ligature Formation | 39 |
| 3.2.1.8 | Strokes | 39 |
| 3.2.1.9 | Script..... | 39 |
| 3.3 | Nastaleeq Script..... | 40 |
| 3.3.1 | Properties of Nastaleeq Script..... | 41 |
| 3.4 | Noori Nastaleeq | 47 |
| 4 | Problem Statement..... | 48 |
| 4.1 | Problem Scope | 49 |
| 5 | Methodology | 50 |
| 5.1 | Scan Image / Open Image..... | 53 |
| 5.2 | Separate Line of Text..... | 54 |
| 5.3 | Set Base Line | 55 |
| 5.4 | Isolate Ligature and Diacritics..... | 55 |
| 5.5 | Segmentation of Main body..... | 55 |
| 5.6 | Framing..... | 60 |
| 5.7 | Hidden Markov Model..... | 60 |
| 5.7.1 | HMM Training..... | 61 |
| 5.7.2 | HMM Recognition..... | 62 |
| 5.8 | Recognizing the Ligature..... | 62 |
| 6 | Results..... | 65 |
| 6.1 | Purpose..... | 65 |
| 6.2 | Sample | 66 |
| 6.3 | Method | 66 |
| 6.4 | Test Results..... | 66 |
| 6.4.1 | Character level Results | 66 |
| 6.4.1.1 | Class of Alif..... | 66 |
| 6.4.1.2 | Class of Swad..... | 66 |
| 6.4.1.3 | Class of Dal..... | 67 |
| 6.4.1.4 | Class of Choti Yeh..... | 67 |
| 6.4.1.5 | Class of Ain | 67 |
| 6.4.1.6 | Class of Bay | 67 |
| 6.4.1.7 | Cumulative Results..... | 67 |
| 6.4.2 | Ligature Level Results | 68 |
| 6.5 | Examples..... | 68 |
| 7 | Discussion..... | 70 |
| 7.1 | Distortion in the Image | 70 |
| 7.1.1 | Problem Description | 70 |
| 7.1.2 | Example | 70 |
| 7.1.3 | Solution..... | 71 |
| 7.2 | Similarity in Shape..... | 71 |
| 7.2.1 | Problem Description | 71 |
| 7.2.2 | Examples..... | 71 |
| 7.2.3 | Solution..... | 72 |

| | | |
|-------|-----------------------------------|----|
| 7.3 | Inconsistency in Font | 72 |
| 7.3.1 | Problem Description | 72 |
| 7.3.2 | Example | 72 |
| 7.3.3 | Solution..... | 73 |
| 8 | Conclusion | 73 |
| 9 | Future Work and Enhancement | 74 |
| | Reference | 75 |
| | Appendix..... | 80 |
| | Appendix A..... | 80 |
| | Appendix B..... | 88 |

1 OCR

Optical character recognition, usually abbreviated to OCR, is computer software that is designed to translate text images into machine-editable text document that can be opened in any word processor or text editor. It helps to quickly digitize paper documents for further manipulations without any manual effort. OCR began as a field of research in pattern recognition, artificial intelligence and machine vision. [35].

1.1 The Need for OCR Systems

OCR (Optical Character Recognition) allows us to manipulate the printed data, using computer with minimum effort and time. It converts the scanned image to text-based document, which can be easily processed by a word processor or text editor.

Consider the benefits [51]:

- **OCR saves space.** Bulky files, which require a lot of space, can be easily replaced by small sized digital documents. It saves acres of space that has been once given to paper cabinets and boxes.
- **OCR saves time.** No need of time consuming retyping process. Not only the data manipulation but also the information retrieval becomes very easy.
- **OCR saves efforts.** The text can be reused, edited or reformatted effortlessly.
- **OCR saves worry.** Digital backup copies of crucial documents can be maintained which reduces the probability of error.

1.2 Generic Model of an OCR System

One of the main characteristics of Optical Character Recognition is to study automatic reading. The most impressive and astonishing capability of human brain is to recognize patterns in nature. Therefore, the aim of OCR is to emulate the human ability to read at a much faster rate by associating symbolic identities with images of character.

The language independent text recognition process can be broadly classified into three main categories, which are:

- Image Acquisition.
- OCR Software.

- Output Interface.

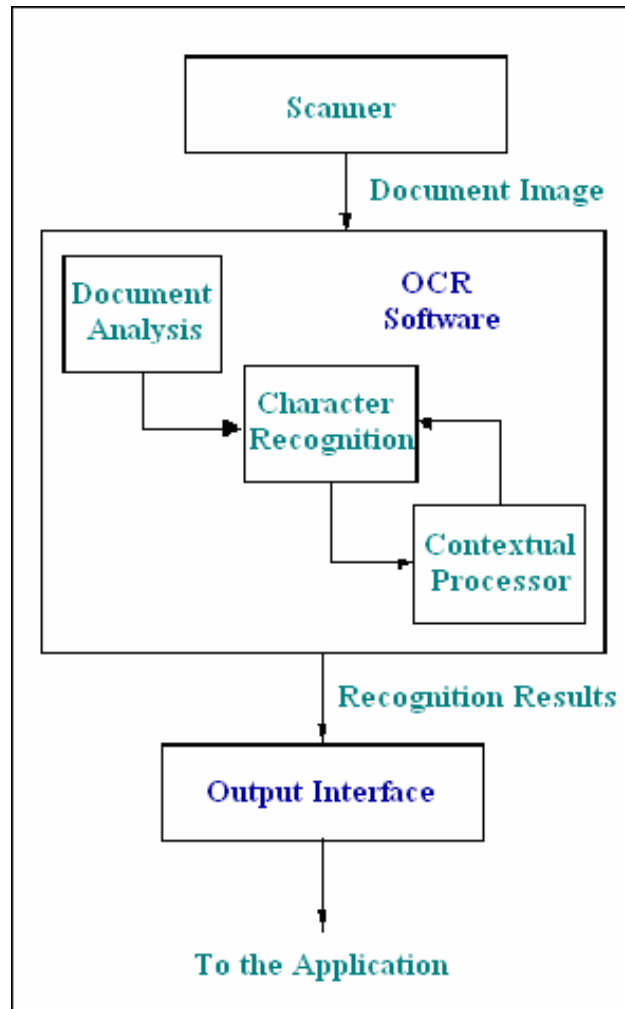


Figure 1 : Basic Model for OCR

1.2.1 Image Acquisition

In the very first phase of OCR we need to acquire an image, which needs to be converted. Scanners are normally used to acquire an image. In some cases digital cameras are also used.

1.2.2 OCR Software

This is the main step of OCR where the actual translation of the text image to machine editable text is performed. The characters from the preprocessed image are recognized and are converted to text. This process comprises of three main operations.

1.2.2.1 Document Analysis

It is also called pre-processing process in which text is extracted from the document image. This operation is very important for the better results.

Different steps involved in the Pre-Processing phase are as follows

1.2.2.1.1 Skew Detection and Removal

Skew is the distortion or tilt, by an angle, in the input image i.e. the angle of the baseline that is not written horizontally. Sometimes it so happens that while scanning a page using a scanner, the page is tilted to a certain angle. As a result whole text is tilted and the angle of the line makes it inappropriate for OCR processing. So we may say that skew detection and removal plays a significant role in output of OCR project [47].

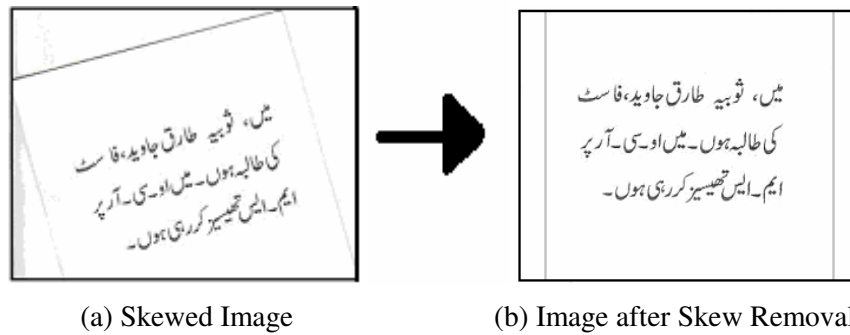


Figure 2 : Skew Detection and Removal

1.2.2.1.2 Binarization

The binarization is a process, which converts a gray scale image to a black and white image. The simplest way to do this is through thresholding, in which a histogram of the grey values of an image is computed and the cut-off point (valley between the two peaks) is calculated. All the pixels whose value is above the cut-off point are converted to one while all others to zero [47].

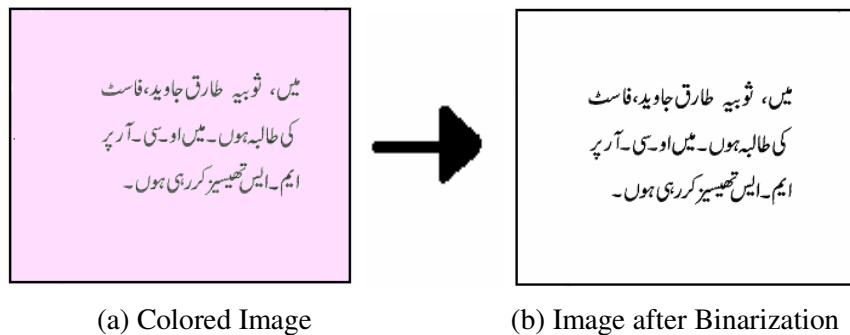


Figure 3 : Binarization

1.2.2.1.3 Filtering and Smoothing

'Filtering and Smoothing' is basically done to remove noise and distortion from the image, which may be produced in image acquisition. The filtering process is used to remove distortion and noise produced at the time of scanning due to shot noise, dark current noise, thermal noise or cross-coupling noise. Fine-textured noise is removed through smoothing [47].

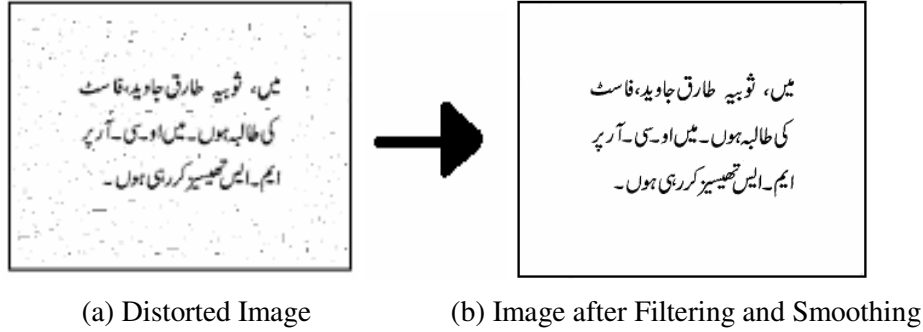


Figure 4 : Filtering and Smoothing

1.2.2.1.4 Thinning

The thinning is a morphological process, which is an efficient way to express the structural relationships in the character recognition as it removes the selected foreground pixels from the image. It reduces the computational time and effort to traverse an image. But the technique is very much sensitive to noise; a little disturbance can change the shape of an image.

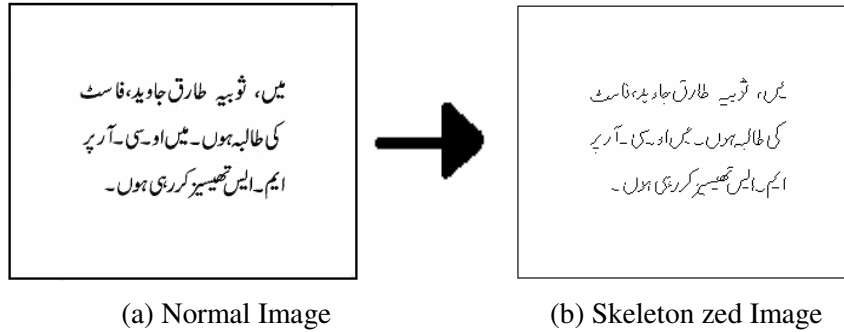
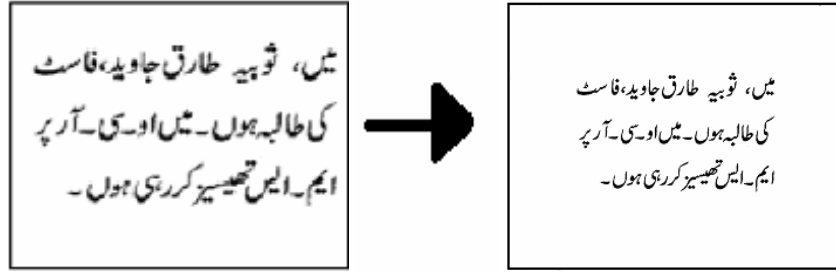


Figure 5 : Thinning

1.2.2.1.5 Normalization

Normalization is generally done to overcome size and orientation variation problem. Our purpose is to correctly recognize the similar shapes with different sizes and orientations. The translation, rotation and scaling is performed prior to the start of the actual OCR processing, to normalize the text for size.



(a) Image before Normalization

(b) Image after Normalization

Figure 6 : Normalization

All these steps are performed to enhance the image for better recognition.

1.2.2.2 Character Recognition

This is the main process where text from the image is read and translated into a form that the computer can manipulate. Various techniques are applied to correctly recognize the text from the image. The features are extracted from the images by the most common method of character recognition, called “feature extraction” and then its shape is analyzed and its features are compared against a set of rules that distinguishes each character. This analysis and comparison identify the characters.

1.2.2.3 Contextual Processing

Results of recognition can be significantly improved by using the contextual information. Higher level of information, which is not available to the classifier, is used to verify the accuracy of the solutions returned by the classifier. One of the methods for post processing is to apply spell checker on the classified output to correct the misspelled words due to wrong classification [47].

2 Literature Review

As Urdu (Noori Nastaleeq) is a cursive script, so most of the literature provided in this document is either related to Urdu offline Handwritten OCR or Cursive Handwritten Latin OCR.

2.1 Background and History

The attempts to automate printed material, for further data manipulation, started prior to World War II. The need of OCR was first realized in banks for automatic form checking. The modern OCR technology started with the invention of GISMO-a Robot by M. Sheppard in 1951. In 1954, J. Rainbow developed a prototype machine, which was able to read uppercase typewritten output at the incredible speed of one character per minute [32].

Then a series of dramatic developments took place in technology, during the late 1960s. Then in 1971, the OCR technology revolutionized bill payment systems and many others [32].

Very fast, reliable, fonts independent and less expensive with higher accuracy OCR systems are available today.

2.2 Research Techniques for Character Recognition

This process can be further divided into two steps.

2.2.1 Segmentation

The accuracy of recognition process is heavily dependent on the type of the word decomposition technique used because the wrong segmentation produces misrecognition or rejection. For recognizing cursive script, there are two main approaches to deal with connected characters in a word.

2.2.1.1 Segmentation Free Approach

In segmentation free approach, the ligature as a whole is used instead of segmenting it into smaller units. The above approach has been used with different pattern matching techniques, which focuses on using features of image as a whole to classify the word, for better performance.

One approach is to use line as a whole instead of further segmenting it into ligatures or words. This technique is mainly applied with HMM where language independent global transformation features are extracted from line and fed to HMM for recognition [5], [6], [7], [18].

In 1994, Elms, A.J [20] used segmentation free and HMM based, level building recognizer, for the recognition of connected characters in which whole connected body is considered as one segment. Though this method can work effectively for the Latin Script but it will not produce fruitful results for Urdu script, which has main bodies as well as diacritics.



Figure 7 : Connected Component Method

Sometimes a combination of connected component labeling and centroid-to-centroid distance method is used to extract ligatures from a line [26]. First a connected component labeling is

applied to separate each connected body irrespective of main body or diacritic. Then centroid-to-centroid distance method is applied to associate each main body with its diacritics.



Figure 8 : Connected Component and Centroid-to-centroid Distance Method

Another effective method is to consider all the connected components touching along the baseline as a ligature. Zahra Shah and Farah Saleem [22] used this method to separate main bodies from diacritics.

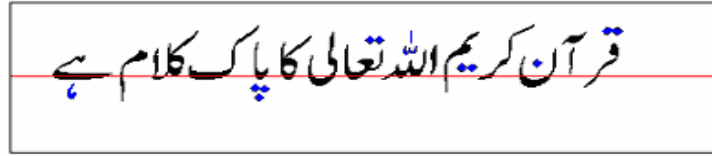


Figure 9 : Connected Component and Baseline Method

Much of the recognition by expert system was done using segmentation free approach. The Generalized Hough Transformation is applied on the image to detect objects in the image [13].

Most commonly used method for Latin script is to calculate vertical projection profile of each line and to segment a line where the histogram has a zero value. That means a contour between two gaps in the line is considered as a ligature. But again this method cannot be applied to Nasteleeq script where the ligatures overlap.

Though a segmentation free approach has proved remarkable for the recognition process and diminished the segmentation and computational effort, but it has also made the recognition process more ligatures dependent in certain cases.

2.2.1.2 Segmentation Approach

The performance of any Arabic text recognition depends on how successful the system overcomes the obstacles of cursive ness and context-sensitivity. The conventional approach is to segment the words into either characters or symbols. Therefore, segmentation based approach has further two divisions.

2.2.1.2.1 Segmenting into Characters

The ligature is correctly segmented into characters. In this approach segmenting accurately is the most crucial part of an OCR. In segmenting the ligatures into characters, such techniques are applied that prevent breaking up of character into more than one part. One approach is to segment a sub-word by applying three sets of rules for the different positions in a sub-word: the first, the last and the middle [23]. Another effective segmentation method is to search for the connection points along the base line. One of the most common methods of segmentation uses the vertical projection histogram and defines the connection points to be the locations where the value of the threshold dips below a certain value.



Figure 10 : Histogram and Connection Point Method

Roberto J.Rodrigues and Antonio Carlos Gay Thome [31] used projection histogram for the initial character segmentation and then further refined until a satisfactory performance was reached. In [33] the combination of component labeling and vertical profile methods was used to segment characters. First vertical projection profile method was applied on the connected characters and then component labeling was applied to correct the mis-segmentation if any. M. Mohamed and P. Gader [2] stated that combining segmentation-free and segmentation-based techniques would give better results. Few segmented the word based on the special features of the characters like slopes, peaks etc.

2.2.1.2.2 Segmenting into Primitives

Segmenting a word into parts smaller than a character called symbols i.e. segmentation at all potential connection points and then combine them to make a character. This means segmenting the sub-word into primitives possibly smaller than a character, like strokes, intersection points and loops etc. This technique is a precondition for feature analysis [4].

Words are segmented into primitives based upon each pixel neighborhood. In 1995, H. Bunke, M. Roth and E. C. Schukat-Talamazzini [17] proposed that the skeleton of a word can be broken down at the pixel, with either one or more than three neighbors, to get loops and edges.

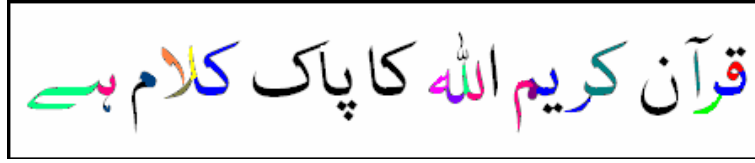


Figure 11 : Neighborhood Method

M. Mohamed and P. Gader [2] segmented the word image into primitives and then used dynamic programming to find the best sequence of segments to match a given string.

Badr Al-Badr and Robert M. Haralick [11] described symbols in terms of shape primitives to recognize machine-printed Arabic words without prior segmentation.

Another method is to cut the connected component into smaller parts called frames. In 1997, A. Kornai [3] favored the character-level pre-segmentation due to the fact that sliding window feature extraction is on average 10% worse. But it will be wrong to make such a conclusion on this 10% difference because this minor difference may be due to the relatively little cursive nature of data set.

Similarly few primitives can be extracted based on the special features of the characters. A 1999 system [4] by Khorsheed and Clocksin segmented the skeletonized script at end points, branch point and cross points. Then these segments were fed to the HMM recognizer, in the descending order relative to the horizontal value of the starting feature point, for the final recognition.



Figure 12 : Edge Based Segmentation Method

Another effective segmentation method is to search for the connection points along the base line. The mixture of neighborhood and baseline method can also effectively segment the skeleton into primitives [34]. The segments can then be recognized by a structure analysis approach.

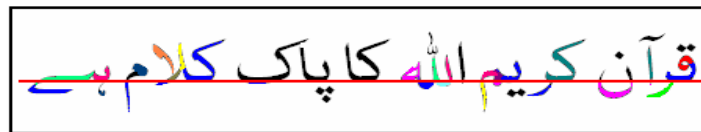


Figure 13 : Neighborhood and Baseline Method

This approach has an advantage over the full character segmentation, as it is easier to find a set of potential connection points, than to find the actual connection points directly.

2.2.2 Recognition

The character recognition algorithm can be sub divided in to two main components – The Feature Extractor and the Classifier.

The descriptors, or feature set, used to describe all characters in a given character image are analyzed and determined by the Feature Extractor which are then used as input to the character classifier. The classifiers basically perform pattern-matching task.

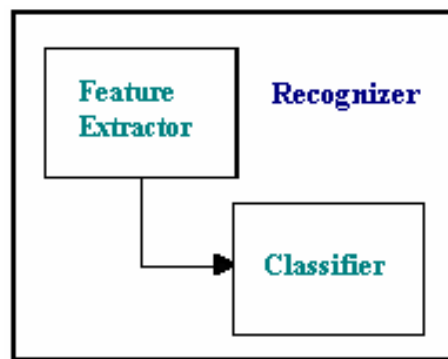


Figure 14 : Character Recognizer

2.2.2.1 Feature Extraction

Simplification of the features required to describe a large set of complex data accurately and efficiently is the main task of feature extraction. The analysis of complex data requires a large amount of memory and computation power or a classification algorithm, which over fits the training sample and generalizes poorly to new samples. Therefore, we can say that feature extraction is a term used for methods of constructing combinations of the variables to overcome these problems well still describing the data with sufficient accuracy [35].

Once the pattern is acquired, the next step is to extract the features of the pattern and pass them to the classifier for recognition. The process of pattern recognition is heavily dependent on feature extraction. The small feature set, which can efficiently discriminate between patterns of different classes, should be selected. One should make sure that these features should be similar for patterns within the same class [47].

The two main control approaches for feature extraction and classification are:

- **Interleaved Control:** In such systems, the pattern is classified incrementally i.e. few features are extracted from a pattern and then a pattern is tried to be recognized in number of passes with the addition of new features in each pass. So it's the alteration between feature extraction and classification [47].

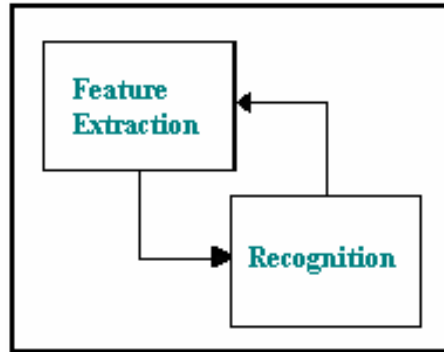


Figure 15 : Interleaved Control

- **One-Step Control:** In One-Step control, first all the required features from a pattern are extracted and then the pattern is classified [47].

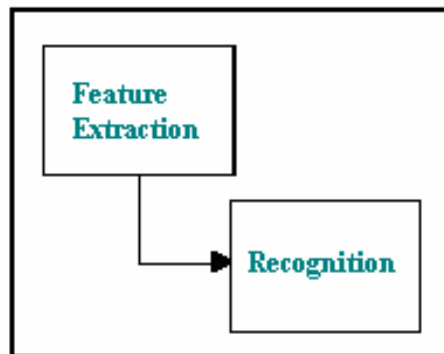


Figure 16 : One-Step Control

There are three different types of features, which are:

2.2.2.1.1 Structural Features

Structural features describe a pattern in terms of its structure and geometry by giving its global and local properties. The pattern is carefully analyzed to extract these features from it. Thus it may be said that these features are heavily dependent on the kind of the pattern to be classified. In

the case of characters, the features include strokes, and bays in various directions, end points, intersection points, loops, dots and zigzags [47].

A pseudo- neuronal system [29] for reading cursive script extracted ascender, decender and loops from the image. A 1999 system [4] by Khorsheed and Clocksin extracted normalized length, curvatures, endpoints, relative location and curved features which include the number of pixels above the top feature point, below the bottom feature point, left of the left most feature point and right of the right most feature point of that segment, from a word's skeleton for recognition. In a template matching approach [22], features like location of dots and placement of other diacritics for each ligature are extracted. In 2001, M. Fahmy and S. Al Ali [25] used features like junctions, turning points, loops, strokes etc and fed them to neural networks to automatically recognize off-line handwritten Arabic words and obtained a recognition rate of 69.7%. Then in 2002, S. Snoussi Maddouri, H. Amiri, A. Belaïd and Ch. Choisy[27] used a combination of global (baseline, ascenders, descenders, positions of primitives and loops) and local features (position normalization, starting point normalization, harmonic phase normalization and size normalization) for recognition.

Some approaches use the height, width, number of cross-points, and the category of the pattern (character body, dot, etc), the presence and number of dots and their position with respect to the baseline, the number of concavities in the four major directions, the number of holes and the state of several key pixels, the number of strokes or radicals and the size of the stroke's frame, and the connectivity of the character.

In 2000, D. Megherbi, S.M. Lodhi, J.A. Boulenouar [12] used Fuzzy logic to make a classification of 36 Urdu characters into seven sub-classes, namely sub-classes characterized by seven well defined fuzzy features (number of dot(s) present in the character, place of the dot, branch or presence of secondary stroke, ratio (height to width), slope between initial point and final point, initial angle and begins with intersection) to characterize Urdu characters. H. Bunke, M. Roth and E. C. Schukat-Talamazzini [17] used the feature set, having information related to spatial location of an edge, curvature and percentage of pixels around the curved edges for offline cursive handwriting recognition. In [8] a single hidden Markov model (HMM) was trained with the structural features (dots, end points, branch points, cross points, simple loop, complex loop and turning points) extracted from the words.

Sometimes features based on the concept of topological, contour and water reservoir are extracted [33].

Extraction of structural features from an image is not an easy task but it can very well accommodate distortions and variations in writing styles [47].

2.2.2.1.2 Statistical Features

The statistical features describe a pattern in terms of a set of characteristics measurements (median, stand deviation, variance, ratios etc) extracted from the pattern. That is to say the features based on the statistics derived from the pattern. These features normally require some kind of computation or calculation. Zoning, characteristic loci, crossings and moments are the statistical features that are mainly used for Arabic text recognition [47]. A multi-tier holistic approach [26] for the offline recognition of cursive Urdu Text (Noori Nastaliq Script) used measurements like *solidity, Axes Ratio, Eccentricity; Moment based features, normalized length feature, and curvature feature and number of holes*.

Sometimes Local Line Fitting is used as features by fitting the pixels belonging to each receptive field to a straight-line using eigenvector line fitting or orthogonal regressions [10].

Sometimes each column in a word is represented as a feature vector (transition features) [2].

In general, statistical features can be easily and rapidly extracted from a text image. Apart from their tolerance to accommodate moderate noise and variation, they also make systems robust for new fonts [47].

2.2.2.1.3 Global Transformations

By the transformation scheme, the pixel representation of the pattern is converted to an alternative more abstract representation. As a result, the dimensionality of features is reduced [47].

One approach is to use a transformation based on vertical and horizontal projections of a pattern i.e. length of the ON pixels in columns or rows respectively. The comparison of each row or column can further add to the information. [20], [1].

This scheme is basically used with HMMs. Sometimes same advanced technology that is used for speech recognition is utilized that is to compute Intensity (percentage of black pixels within each cell) as a function of vertical position; Vertical derivative of intensity (across vertical cells); Horizontal derivative of intensity (across overlapping frames); Local slope and correlation across a window of 2-cell square for each frame of image and pass it as a feature vector to HMM[5].

In another approach pixel values are used as the basic global transformational features but this approach results in a vast amount of features for each frame. In order to reduce the number of features a Loeve-Karhunen Transformation is performed on the grey values of each frame [7].

Unlike statistical and structural features, transformation schemes can be easily applied. They can tolerate noise and variation but sometimes require the use of additional features to obtain high recognition rates [47].

2.2.2.2 Pattern Matching / Classification

The classification is the main decision making stage in OCR. Classifier tries to classify the pattern, as a member of a certain class, by comparing the extracted features to those of the model set. Classification does not produce a unique solution instead give a set of approximate solutions as an output [47].

Some widely used recognition techniques are discussed below.

2.2.2.2.1 Template Matching

Template matching is a pixel-by-pixel comparison of a pattern with a set of pre-defined pattern templates. It is considered that the pattern, under consideration, belongs to the class of the template to which it is most similar. Each extracted word from the image is compared with a predefined shape words from a predefined data set. The basic method is to go through all the pixels in the image and compare them to the pre-defined templates. After all templates have been compared with the image, the unknown image's identity is assigned as the identity of the most similar template. Though this method is the most simple and easy to implement but it is very slow in process [47]. In 1998, Syed S. Hyder and Ali Khoujah [23] used template matching to classify the word. They first determined the different shapes (final, medial, isolated and initial) of each letter in the printed cursive script and then developed context sensitive grammar to accept a shape by pattern matching with sub-shapes or templates that constituted a class of character shapes.

Zahra Shah and Farah Saleem [22] used template matching for the recognition of type written Urdu Nastaleeq script and obtained an accuracy rate of 79%. The *two pass method for the recognition* of visible features was used. The diacritics were recognized in the first pass by template matching and then the visible features recognition strategy was applied on the main bodies to correct any misrecognition of diacritic as a main body.

There are several disadvantages of using template matching; firstly the limited number of fonts can be recognized using the template matching. Secondly, a little variation in the image can

greatly affect the output of the OCR. Thirdly, it requires great computational effort and time for recognition.

2.2.2.2.2 Statistical Classifier

The statistical pattern recognition is a machine intelligence approach, which generates stochastic parameters to characterize the properties of the pattern to be recognized. Its main purpose is to determine the class or category for a given sample. This is achieved by using statistical methodologies such as statistical hypothesis testing, correlation and Bayes classification [32].

2.2.2.2.2.1 Neural Network

In recent years, neural network have been successfully applied in many tasks of pattern recognition and machine learning systems. An Artificial Neural Network (ANN), information processing paradigm, consists of larger number of highly interconnected processing elements called neurons, whose structure is drawn from analogies with biological neural systems, working in union to solve a specific problem. ANN, like people, learns by example [50].

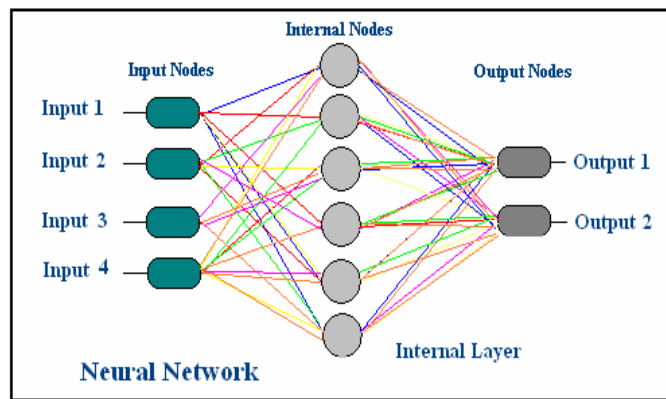


Figure 17 : Neural Network

In 1995, Myriam Côté, Eric Lecolinet, Mohamed Cheriet and Ching Y. Suen [29] recommended pseudo- neuronal system for reading cursive script. The system was based on hierarchically organized three levels of detectors: *the feature detectors, the letter detectors and the word detectors*. Interaction between the detectors was through excitatory activation of neural type. The system was in fact modified form of perception model of McClelland & Rumelhart with addition of some characteristics specific to cursive script.

J. Hébert, M. Parizeau and N. Ghazzali [24] used Neural Networks for isolating handwritten characters in cursive words without prior segmentation and without applying any constrain. The

key idea behind this strategy was to move a window of attention around the cursive word, search for instances of known characters. If the current window contains some significant part of a character, then translate and scale the window of attention in such a way that it converges to the bounding box of that character. However, this approach can be completely trained using data sets of isolated characters only.

In 2001, M. Fahmy and S. Al Ali [25] used geometrical features and neural networks to automatically recognize off-line handwritten Arabic words and obtained a recognition rate of 69.7%.

Then in 2002, S. Snoussi Maddouri, H. Amiri, A. Belaïd and Ch. Choisy[27] used the idea of the preceptor system, developed by M. Cote for Latin word recognition, for the recognition of Arabic hand-written words. He combined a global and a local vision modeling (GVM- LVM) of the word in a NN called Transparent Neural Network. A recognition rate of 97% was achieved. Text was recognized first by examining the words and if that failed then the individual letters were examined.

Again in 2002, Syed Afaq Husain and Syed Hassan Amin [26] proposed a new multi-tier holistic approach for the offline recognition of cursive Urdu Text (Noori Nastaliq Script). The special ligatures (Dot, Tay, Hamza and Mad) were identified first from the base ligatures using a Feed Forward Back propagation neural network with 15 inputs, 25 hidden and 25 output neurons. The feature vectors were fed to this neural network. It then identified the ligatures as either special ligatures or base ligatures. In the second step, these special ligatures (diacritics) were associated to the main bodies by using minimum *centroid-to-centroid distance*. Finally, the extracted information (main bodies along with their diacritics) was fed to the Feed Forward Back Propagation neutral network for the final recognition task. The network architecture consisted of 34 inputs, 65 hidden neurons and 45 output neurons. The performance of the system on images containing the trained ligatures only was 100%. However untrained additional ligatures were classified to the closet match in the training set since no rejection class was utilized. The main disadvantage of this strategy was that only trained data set would be recognized properly.

A *Back propagation neural network*, which is a multilayer perception model with an input layer, one or more hidden layers and an output layer is used in simple OCR application. Alex Cherkasov [28] presented a technique for the development of OCR using *Artificial Neural Network*.

Though recognition by NN has remarkable contribution to the development of OCR, but training of neural networks is a difficult task, moreover the network created by NN is very complicated. A mesh kind of network is created which is difficult to learn and understand.

2.2.2.2.2 HMM

A hidden Markov model, commonly abbreviated as HMM, is a statistical model where the system being modeled is assumed to be a Markov process with unknown parameters. Based on this assumption, the hidden parameters are determined from the observable parameters. These extracted model parameters are used in the pattern recognition applications for further analysis. The pattern may be a speech pattern or an image pattern [48].

The Hidden Markov Model has a finite set of states where each state is associated with another through some probability distribution. Transitions among the states are governed by a set of probabilities called state-transition probabilities. At a particular state, the outcome or observation is generated according to the associated probability distribution called an observation symbol probability. The state is not visible but only the outcome of the state can be seen to an external observer, therefore states are "hidden" to the outside; hence the named Hidden Markov Model. Since two types of probabilities govern the movement from one state to another so the system is doubly stochastic in nature [48].

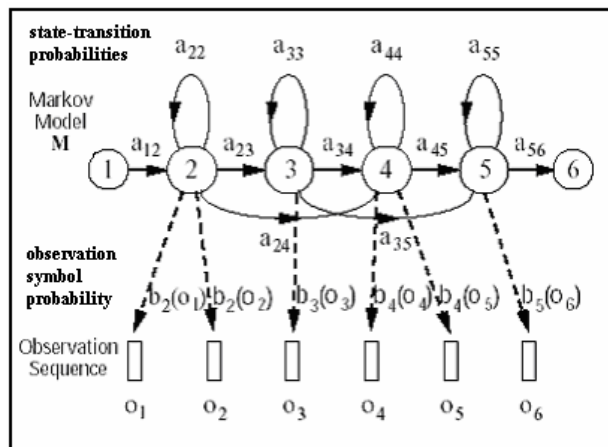


Figure 18 : Hidden Markov Model [49]

Continuous Hidden Markov Model and level building dynamic programming algorithm [19] were used for the recognition of connected and degraded characters. Using a structural analysis algorithm, a number of competing HMMs were used to recognize characters, derived from the small segments of each word.

Elms, A.J [20] proposed a character recognition technique without a prior segmentation in which global transformational features (horizontal profile) were fed to HMMs for recognition. The level building recognizer provided the internal segmentation as well as recognition. The recognition rate of 97.8% was achieved but the character pairs (pb) and (dq) still caused problems.

A.J. Elms along with J. Illingworth [1] again used global transformational features and HMMs for the recognition of isolated characters and claimed a performance of 94%. They represented the character by analyzing the pixel pattern in columns/rows of its image and linked sequential column/rows patterns together with a HMM. Shift Invariant Hamming Distance and the center of gravity of each column/row were used as a feature vector. Since the horizontal and vertical shape profile requires significant computational effort and some pre-segmentation at character level, so the system becomes inappropriate for the languages with connected script.

In 1995, H. Bunke, M. Roth and E. C. Schukat-Talamazzini [17] came forward with the off-line recognition technique for cursive handwriting based on Hidden Markov Model. The features were derived from the arcs of skeleton graphs of the word and fed to HMM for recognition. The recognition rate of 98% has been achieved in experiments with cooperative writers using two dictionaries of 150 words each.

An MD-HMM approach (one-model per word) is called Non-Ergodic HMM (NEHMM). Non-Ergodic HMM grows linearly with the dictionary size and is computationally very expensive. Amlan Kundu, Yang He and Mou-Yen Chen[21] came with the idea that if the computed duration statistics from VDHMM is used to implement MD-HMM approach, then the experimental results will improve significantly.

A 1999 system [4] by Khorsheed and Clocksin extracted structural features from a word's skeleton for recognition, and converted the skeletonised script into an observation sequence suitable for an HMM recognizer. Though recognition rates of up to 97% were achieved but this technique still requires a considerable computation effort and much better preprocessing. Moreover, this technique does not suit Arabic handwritten words, since diacritics are not written exactly below or above each character or edge feature as described in the paper. Also, some of the geometrical features don't suit handwritten Arabic words [16].

Zhidong Lu, Issam Bazzi, Andras Kornai, John Makhoul [5] , suggested that image recognition will be the same as speech recognition in which global transformational features will be calculated for each frame of the window along the image and will be passed as feature vectors to

HMM for final recognition. The system only requires a sample retraining along with ground truth for each new language or script.

Marija Bojovic & Milan D. Savic claimed that systems based on discrete HMMs are generally much faster than systems based on semi-continuous HMMs.[6]

Simon Günter and Horst Bunke [15] proposed a new combination method for HMM based handwritten word recognizers. They combined various HMMs at a more elementary level. The sum of time complexities of the recognition process of the individual classifiers was much lower than the time complexity of the above mentioned technique. Though this method produced favorable results but the overall high time complexity of the recognition process made the system inappropriate.

Somaya Alma'adeed, Colin Higgens, and Dave Elliman [16] presented a complete scheme for totally unconstrained Arabic handwritten word recognition based on a Model discriminant HMM. After applying pre-processing and skeletonization, the classification process based on the HMM approach was used. The output was a word in the dictionary. A detailed experiment was carried out and recognition rate of 45% was reported, which is very low rate.

There is some work done on the offline recognition system for Arabic handwritten words. The recognition system of *Mario Pechwitz and Volker Maergner* was based on a semi-continuous 1-dimensional HMM. Sliding window approach was used to collect the features and testing was performed using the new IFN/ENIT - database of 26459 handwritten Arabic words (Tunisian town/village names) by 411 different writers. On the word level the recognition rate was 89%[7].

In another method [8] features were extracted from the manuscript words and trained a single hidden Markov model (HMM). The HMM was composed of multiple character models where each model represented one letter from the alphabet. Testing was done using samples extracted from a historical handwritten manuscript.[8]

R. El-Hajj, L. Likforman-Sulem and C. Mokbel [18] used continuous 1D HMM for the offline-handwritten recognition. Language independent features, such as lower and upper baselines, were extracted from the image. Thus the word variability due to lower and upper parts of words was tackled properly. Internal segmentation was applied by the continuous 1D HMM. Initially the recognition rate was 74.90% but it improved later on to 86.51% with the addition of the features related to the detected baselines. This technique cannot be applied to Nastaleeq style, a cursive writing style, where there is no specific baseline.

2.2.2.2.3 Clustering

Clustering is the process of organizing objects into groups whose members are similar in some way. i.e. the objects within a same cluster are similar while they are dissimilar to the objects belonging to other clusters. New features can be constructed, which are the abstractions of the existing features, by using these clustering algorithms. Some algorithms, like k-means, simply partition the feature space. Other algorithms, like single-link agglomeration, create nested partitioning which form taxonomy [50].

Fuzzy logic is a problem-solving control system methodology, which is derived from fuzzy set theory dealing with reasoning that is approximate rather than exactly deduced from classical predicate logic. That is to say, it provides a remarkably simple way to draw definite conclusions based upon imprecise, ambiguous, vague, fuzzy, blurred, noisy, or missing input information. It mimics the human ability to make decisions. Fuzzy reasoning involves three steps *Fuzzification* of the terms that appear in the conditions of rules, *Inference* from fuzzy rules, *Defuzification* of the fuzzy terms that appear in the conclusions of rules [12].

In 2000, D. Megherbi, S.M. Lodhi, J.A. Boulenouar [12] used Fuzzy logic to make a classification of 36 Urdu characters into seven sub-classes. Fuzzy logic is used, as it is viewed as generalization of multi-valued logic. All kind of uncertainty and imprecision in knowledge representation, inference, and decision analysis would be dealt properly and easily with fuzzy logic [12].

2.2.2.2.4 Nearest Neighbor

Nearest Neighbor classifier takes a point in vector form. Then computes the distance between this test point and the vector representation of each training set. The training data closest to the test point is considered its nearest neighbor. This helps to allocate a class to the test point [50].

In 2005, Ali Ahmadi, Yoshinori Shirakawa, Md. Anwarul Abedin, Kazuhiro Takemura, Kazuhiro Kamimur, Hans Jurgen Mattausch and Tetsushi Koide [30] proposed an associative memory based system for real-time character recognition. The nearest-distance search algorithm applying the associative memory was used for classification. Though a template matching was used for the classification but mixed analog-digital and fully parallel architecture of the associated memory made the technique very speedy with a small hardware size. The average search time for each character was 45ns.

2.2.2.2.5 Syntactical / Structural

In Syntactical or structural pattern recognition, the pattern representation is through the symbolic data structures such as arrays, strings, trees, or graphs, which portray the relations between fundamental pattern components and also allow hierarchical model representations. Symbolic representation of an unknown pattern is compared with a number of predefined objects for recognition. This comparison is made by a symbolic match so as to figure out the similarity measurement between the unknown input and the number of object models [32].

Very little has been done on machine printed OCR. Badr Al-Badr and Robert M. Haralick [11] used the mathematical morphology operations to detect primitives, from machine-printed Arabic words without prior segmentation, which were then matched with symbol models. The system then yielded the word with the highest posterior probability as a recognized word. The advantage of using this whole word approach versus a segmentation approach is that the result of recognition is optimized with regard to the whole word. The accuracy rate, of preliminary experiments using a lexicon of 42,000 words, was 99.4% for noise-free text and 73% for scanned text.

Stephen Pearce and Maher Ahmed [14] used a powerful graph representation and an evolutionary algorithm framework for segmentation and recognition. First segmentation hypotheses were initialized and then refinements were made to have a fully segmented and recognized string. The evolutionary segmentation was tested in many domains including connected digits, connected characters and simple circuit diagrams. The performance of the evolutionary algorithm depends heavily on the symbol recognition system used.

Roberto J.Rodrigues and Antonio Carlos Gay Thome [31] stated the use of the decision tree algorithm for cursive script recognition based on the use of histogram as a projection profile technique. The basic idea was to construct a tree with successive refinements, from the data of the histograms until an acceptable performance was reached. Successive levels of the tree were allowed based on heuristic criteria. The algorithm included three steps: Image compensation, Initial segmentation and refinement. The results of first stage classifier were quite satisfactory. The use of projection histograms for the process of character segmentation only solved 70% of the problems with connected digits but the problem of trace slant still persisted.

A graph-based structural segmentation approach based on the topological relation between the baseline and the line adjacency graph (LAG) representation of the text was used in [34] . The text was segmented to sub-character units, which were then recognized by a structure analysis

approach. Separate dots and diacritics classifiers were used for their recognition. A regular grammar that described how characters were composed from scripts was used for the final character recognition. The segmentation algorithm failed due to characters touching in irregular positions especially in the case of dot, descender and ascender touching.

The main disadvantage of this approach is its time and computational complexity.

2.2.2.2.6 Expert System

An expert system is a knowledge based computer program that contains some of the subject-specific knowledge/information of one or more human experts.

Eric Lecolinet [9] proposed a model for cursive script recognition, which was based both on a top-down recognition scheme called Backward Matching and a bottom-up feature extraction process. Both worked in a competitive way. Visual Indexes (stroke extraction, loop extraction, baseline determination and VI clustering and normalization) were extracted first by the bottom-up recognition stage. Then, the symbolic descriptions of the words were verified by Backward Matching process. This technique requires pre-processing for better results.

A new representation method for handwritten character recognition called LLF (Local Line Fitting), was proposed by Juan-Carlos Perez, Enrique Vidal and Lourdes Sanchez [10]. They applied geometric operation to extract the features (density and the other features that represent the fitted straight line) from the image. The method yielded a relatively low dimensional and distortion invariant representation. This method can be used by any geometrically based classification systems such as Neural Networks, vector space distance-based methods, discriminant functions, etc as it produced a fixed-size vectors. Later on many handwritten digits and letters recognition applications used this method.

Sofien Touj, Najoua Essoukri Ben Amara and Hamid Amiri [13] proposed an approach for Arabic character recognition based on the use of a Generalized Hough Transform (GHT). Different feature points were extracted and different characters' model was build. A recognition success average rate of 93% was obtained.

U. Pal and Anirban Sarkar [33] used a combination of *topological, contour and water reservoir* concept based features for the recognition of individual characters of Printed Urdu Script. A classifier tree was designed using these features. The characters were recognized in two stages. In the *first stage*, the characters were classified into groups by a feature based tree classifier. In the *second stage*, more sophisticated features were used to recognize similar characters. The *design*

of a tree classifier had three components: (1) a tree skeleton or hierarchical ordering of the class labels, (2) the choice of features at each non-terminal node, and (3) the decision rule at each non-terminal node. It was basically a CART tree. A prototype of the system had been tested on printed Urdu characters and achieved 97.8% character level accuracy on average.

3 Urdu Writing System

Urdu is National Languages of Pakistan. It is from Indo-Iranian subfamily of the Indo-European family of languages. Arabic script in Nastaliq style with several extra characters is used to write Urdu [38]. Many languages share Arabic script of writing, thus written from right to left (RTL) [36]. In Urdu, number system is written form left to right, so Urdu writing system has both the properties of left to right and right to left writing systems as shown in the figure [37]:



Figure 19 : LTR and RTL Directions in Urdu script

The language is not related to Arabic, but to languages of Northern India, especially Hindi [36]. Some people in the Hindustan region (Northern India) embraced Islam during the Mughal period. They started using Arabic script and gave birth to a new language called Urdu. [36]. Others remained Hindu and used the Devanagari alphabet. This language became Hindi [36].

Several letters of the Persian script, which itself is an adaptation of Arabic script, specific to Urdu were added. Urdu script is cursive, the appearance of a letter changes depending on its context/position: isolated, initial (joined on the left), medial (joined on both sides), and final (joined on the right).

Another interesting property of Urdu writing system shows that Urdu writing system is context sensitive, that is, characters change their shapes depending upon the characters following and preceding it as in figure below. This change follows some rules like, shape of the next letter to join with and the shape of the character, which is joining [37].

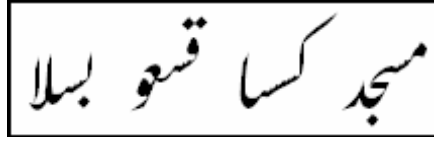


Figure 20 : Context Based Multiple Shapes of س “Seen”

A famous Muslim scholar Ibne-Muqalla developed a new font called Naskh in the year 310 AH. Naskh font dominated all the previously invented fonts due to its easy and fast writing system [37].

One traditional device of writing Urdu is Qalam (chisel-tip of flat-nib bamboo). Size of the flat-nib is primary parameter of size of the shape of an alphabet. In early days, distinguished writers also used feather as writing device. These days, normal rounded-nib writing instruments (namely: fountain-pen, ballpoint-pen, pencil, etc.) are widely used. However, Qalams are also available in the market and used by professional and serious script specialist (termed “Khattaat” in Urdu).

Urdu is written in Pakistan, Middle East, and Muslim parts of India, Mauritius, and South Africa [38]. There are almost 104 million (1999 WA) people (including second language speakers), out of which around 60 million use it as first language. But this data is about Urdu speakers; how many of them can read and write depend on literacy rate and essentiality of Urdu in education.

Following are the primary alphabets along with their IPA symbols and pronunciation (in Urdu and IPA):

| | | | | | | | | | | | | |
|--------------------|------|----|-----|-----|-----|-----|-----|------|------|-----------|-------------|-----------|
| Urdu Alphabets | ا | ب | پ | ت | ٹ | ث | ج | چ | ح | خ | د | ڈ |
| Urdu Pronunciation | الف | بے | پے | تے | ٹے | ثے | جیم | چے | ھے | خے | دال | ڈال |
| IPA Symbols | alif | be | pe | te | te | se | jīm | che | ħe | khe | dāl | dāī |
| Urdu Alphabets | ذ | ر | ڑ | ز | ژ | س | ش | ص | ض | ط | ظ | ع |
| Urdu Pronunciation | ذال | رے | ڑے | زے | ژے | سین | شین | صاد | ضاد | طوئے | ظوئے | عین |
| IPA Symbols | zāl | re | re | ze | ze | sīn | šīn | svād | zvād | toe | zoe | ʿain |
| Urdu Alphabets | غ | ف | ق | ک | گ | ل | م | ن | و | ہ | ھ | ء |
| Urdu Pronunciation | غین | فے | قاف | کاف | گاف | لام | میم | نون | واو | محمولی ہے | دوٹھی ہے | حمزہ |
| IPA Symbols | ğain | fe | qāf | kāf | gāf | lām | mīm | nūn | vāū | choṭī ħe | do ḥasmi ħe | hamzah |
| Urdu Alphabets | | | | | | | | | | | | ی |
| Urdu Pronunciation | | | | | | | | | | | | محمولی ہے |
| IPA Symbols | | | | | | | | | | | | choṭī ye |

Figure 21 : The Primary Alphabets along with their IPA Symbols and Pronunciation

There are some other characters, which are not unanimously considered part of Urdu primary alphabets; they are rather considered variations of the above given alphabets. These disputed characters are listed latter in the table showing Unicode & Glyph Mapping using sub-serial (a) and (b) along with the serial of main primary alphabet.

Hindi/ Urdu together represent the third most spoken language in the world [41]. Urdu has also been written using the Roman script since the days of the British Raj (British Rule in India). [41].

Technically, linguists do not distinguish between Hindi and Urdu as separate languages [35]. It is only the script that distinguishes between the two at a substantial magnitude. Hindi is non-cursive, straight-line, left-to-right, Sanskrit-based script, which is a total contrast to Urdu. This argues the importance of writing systems for identity of language.

Other symbols used in Urdu writing systems are:

Table 1 : Diacritics (Erabs) in Urdu













| | | | |
|-------------|---|-------------|--|
| Zabar |  | Zair |  |
| Pesh |  | Jazm |  |
| Mad |  | Tashdid |  |
| Alif -short |  | Chota toein |  |
| Do zabar |  | Do zair |  |
| Do paish |  | Ulti paish |  |

Table 2 : Urdu Numerals











| | | | |
|-------|---|-------|--|
| Zero |  | Five |  |
| One |  | Six |  |
| Two |  | Seven |  |
| Three |  | Eight |  |
| Four |  | Nine |  |

Table 3 : Punctuation Marks in Urdu

| | | | |
|-----------------|---|-----------------|---|
| Question Mark | ؟ | Comma | ، |
| Semi-colon | ؛ | Colon | : |
| Full stop | . | Dash | - |
| Opening bracket |) | Closing bracket | (|

| | | | |
|-----------------|---|---------------|---|
| Exclaiming Mark | ! | Forward slash | / |
| Open quote | “ | Close quote | ” |

3.1 Character to Unicode and Glyph Mapping

Urdu speakers need both Arabic script support and Urdu support. Arabic code points in the U+0600 - U+06FF range represent all of the letters without regard to their position. It is up to the font to show the letter with the proper appearance.

Table 4 : Different Shapes of Character in Urdu

| Serial # | Character-Name | | Unicode | Alternating Glyphs | Alternating Glyphs with Context (in a ligature*) |
|----------|---------------------------|---------|---------|--------------------|--|
| 0 | Hamzah (without kursi) | ہمزہ | U+0621 | ء | __ء# |
| 1 | Alif | الف | U+0627 | ا | ا_ |
| | (FINAL) | | | ا | _ا |
| 1a | alif madd | الف مدّ | U+0622 | آ | آ_ |
| | (FINAL) | | | آ | _آ |
| 2 | Bē | بے | U+0628 | ب | ب |
| | (INITIAL) | | | ب | ب_ |
| | (MEDIAL) | | | ب | _ب |
| | (FINAL) | | | ب | _ب |
| 3 | Pē | پے | U+067E | پ | پ |
| | (INITIAL) | | | پ | پ_ |
| | (MEDIAL) | | | پ | _پ |
| | (FINAL) | | | پ | _پ |

3.2 Advance Script Properties

Some of the Properties of Urdu Script are:

3.2.1 Character Composition and Decomposition

Few letters can combine together to form a new letter and similarly letters can be decomposed.


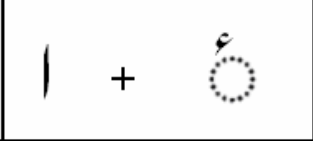
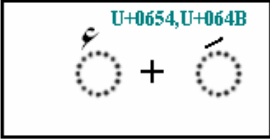
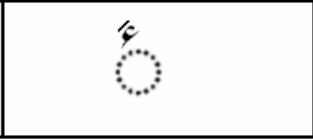
| | |
|--|--|
|  U+0623 |  |
|  U+0654,U+064B |  |

Figure 22 : Character Composition and Decomposition

3.2.1.1 Different Shapes

The Arabic script is cursive, and all primary letters have different forms depending on whether they are at the beginning, middle or end of a word, so they may exhibit 4 distinct forms (initial, medial, final or isolated).

3.2.1.1.1 Isolated Form

If a writing group consists of only one letter, the isolated form is used.

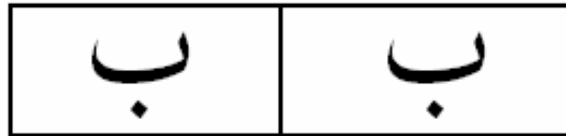


Figure 23 : Isolate form of Bay

3.2.1.1.2 Initial Form

The starting letter of the word takes the initial form i.e. joined on the left.



Figure 24 : Initial form of Bay

3.2.1.1.3 Final Form

The ending letter of the word takes the final form i.e. joined on the right.

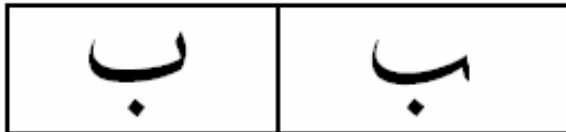


Figure 25 : Final Form of Bay

3.2.1.1.4 Medial Form

The letter, which does not take the initial, final or isolated form takes the medial form i.e. joined on both sides.



Figure 26 : Medial Form of Bay

3.2.1.2 Connection Form

In specified situations, default glyphs are replaced with alternate forms that provide better joining behavior.

| | | | |
|---|---|---|----|
| ب | + | ا | با |
| ب | + | ص | بص |
| ب | + | م | بم |

Figure 27 : Connection Forms

3.2.1.3 Diacritic Placement

For the proper pronunciation of the words, the Urdu characters need some kind of diacritics. The diacritics appear above or below a character to define a vowel or to emphasize a particular sound. There are a number of diacritics, the common ones being Zabar, Zeir, and Pesh.



Figure 28 : the diacritic: Zeir, Zabar and Pesh included with the character say

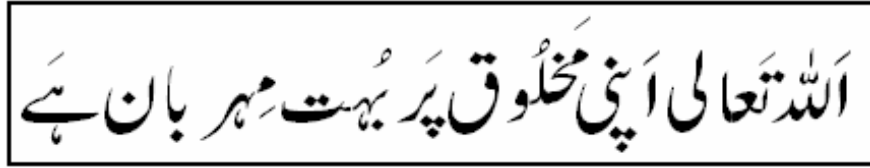


Figure 29 : An Urdu Phrase Written in Nastaleeq Script with Diacritics

3.2.1.4 Presence of Base Line

The base line of Urdu is a horizontal line, which runs through the text, cutting all the words at some point.

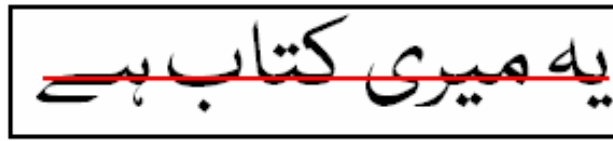


Figure 30 : Base Line in Urdu Text

3.2.1.5 Overlapping

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



Figure 31 : Overlap in the Character

3.2.1.6 Upper and Lower Cases

Unlike English, there is no concept of upper case and lower case in Urdu language writing.

3.2.1.7 Ligature Formation

In Urdu text, more than one character joins together to form ligature. A word may comprise of one or more ligatures and ligature in turn may consist of one or more characters. For example, in the word Pakistan (پاکستان) there are 3 ligatures i.e. Pa (پا), kista (کستا) and n (ن).

3.2.1.8 Strokes

There are two types of strokes of Urdu characters. The main stroke is the longest continuous portion of the character that is written before lifting the pen.

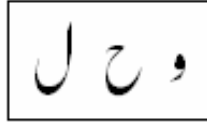


Figure 32 : Main Strokes

The secondary stroke is the other portion of the character written after the main stroke.



Figure 33 : Encircled are the Secondary Strokes

3.2.1.9 Script

Urdu is written using different scripts. Few of them are listed below

| | |
|--------------------------------|---------|
| وَسِحْرَ الشَّمْسِ وَالْقَمَرِ | نستعلیق |
| وَسِحْرَ الشَّمْسِ وَالْقَمَرِ | کوفی |
| وَسِحْرَ الشَّمْسِ وَالْقَمَرِ | ثلث |
| وَسِحْرَ الشَّمْسِ وَالْقَمَرِ | دیوانی |
| وَسِحْرَ الشَّمْسِ وَالْقَمَرِ | رقاع |
| وَسِحْرَ الشَّمْسِ وَالْقَمَرِ | نسخ |

Figure 34 : Examples of Different Urdu Scripts

3.3 Nastaleeq Script

Most widely used Nastaleeq script is taken as a standard for writing Urdu language. It is a combination of two different fonts, Naskh and Taleeq. It was initially created by Mir Ali Tabrezi and has been refined over the past 600 years. It is a complex script as it based not only on the pre-defined rules but also on the aesthetic sense of the calligrapher. It is written using a flat nib (traditionally using bamboo pens) and is highly cursive and context sensitive in nature [44].

Apart from the characteristics of Urdu, Nastaleeq also exhibits some other properties, which make it distinct and complex.



Figure 35 : Nastaleeq Script [37]

3.3.1 Properties of Nastaleeq Script

Some of the intricate features of Nastaleeq are:

- Unlike Naskh, which has only four shapes for each letter, Nastaleeq letters adopt different shapes depending on the context in which it is written [44].
-

Table 5 : Number of Shapes of Nastaleeq Script

| CHARACTER | NUMBER OF SHAPES WITH RESPECT TO POSITION | | | |
|-----------|---|---------|------------|-------|
| | Isolated | Initial | Media 1 | Final |
| Bay | 1 | 23 | 19 | 2 |
| Jeem | 1 | 19 | 17 | 1 |
| Ray | 1 | - | - | 2 |
| Dal | 1 | - | - | 1 |
| Seen | 2 | 18 | 17 | 1 |
| Suad | 1 | 17 | 17 | 1 |
| Toay | 1 | 17 | 17 | 1 |
| Ain | 1 | 17 | 17 | 1 |
| Fay | 1 | 19 | 17 | 1 |
| Kaf | 1 | 19 | 17 | 2 |
| Lam | 1 | 17 | 19 | 1 |
| Meem | 1 | 17 | 17 | 2 |
| Vao | 1 | - | - | 1 |

| | | | | |
|-----------------|---|----|----|---|
| Hey | 1 | 17 | 17 | 2 |
| Hey (do chasmi) | 1 | 17 | 17 | 2 |
| Alif | 1 | - | - | 2 |

Since the Nastaleeq is a highly context sensitive script, a single letter at a particular position may constitute different shapes.

ISOLATED SHAPES

Following are the isolated shapes of some of the characters



Figure 36 : Isolated Shapes [44]

INITIAL SHAPES

Following are the initial shapes of some of the characters.

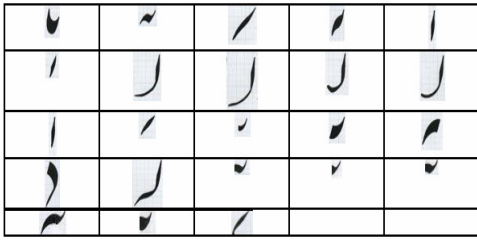


Figure 37: Initial Shapes of Bay [44]

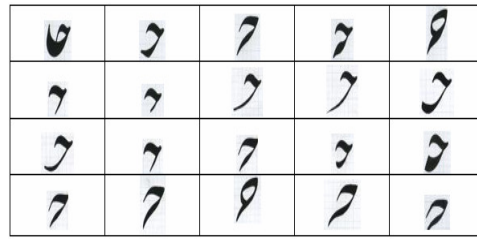


Figure 38: Initial Shapes of Jeem [44]

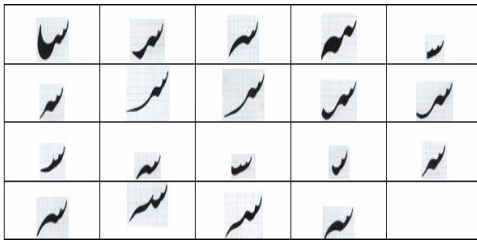


Figure 39: Initial Shapes of Seen [44]

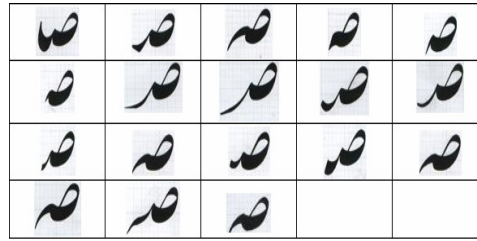


Figure 40: Initial Shapes of Suad [44]

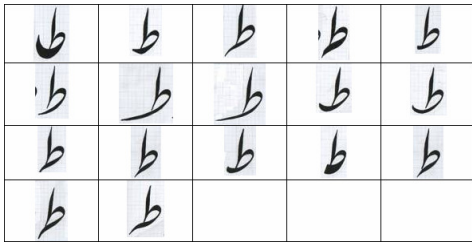


Figure 41: Initial Shapes of Toey [44]

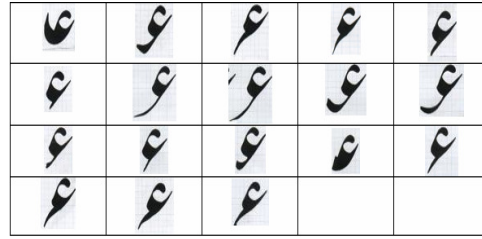


Figure 42: Initial Shapes of Ain [44]

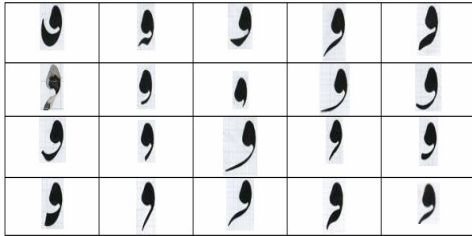


Figure 43: Initial Shapes of Fay [44]

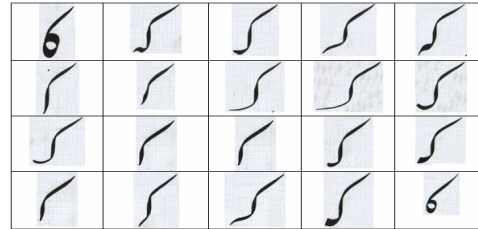


Figure 44: Initial Shapes of Kaf [44]

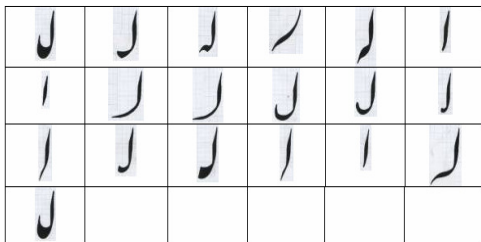


Figure 45: Initial Shapes of Laam [44]

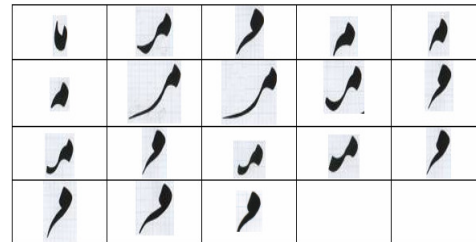


Figure 46: Initial Shapes of Meem [44]

MEDIAL SHAPES

Following are the medial shapes of some of the characters.

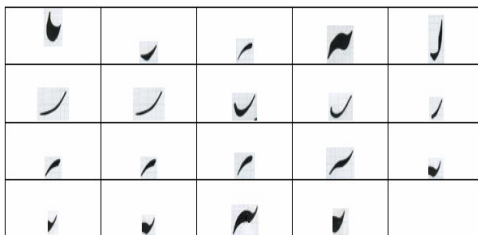


Figure 47: Medial Shapes of Bay [44]

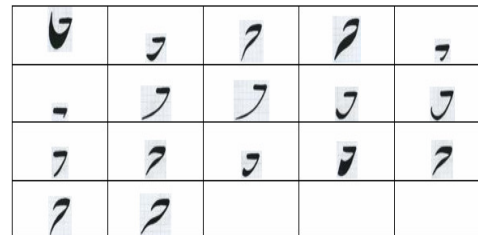


Figure 48: Medial Shapes of Hay [44]

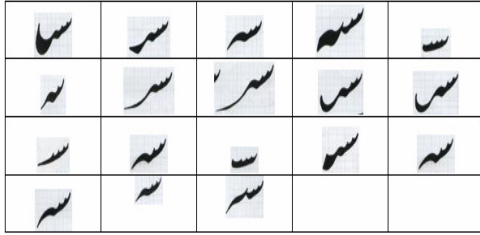


Figure 49: Medial Shapes of Seen [44]

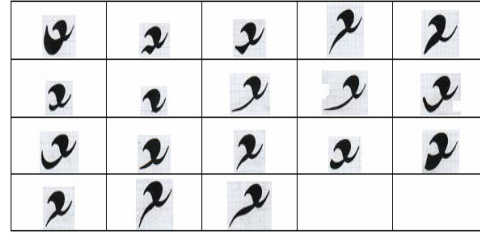


Figure 50: Medial Shapes of Ain [44]

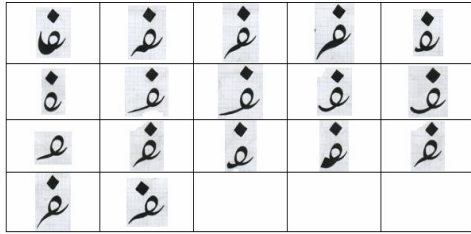


Figure 51: Medial Shapes of Fay [44]

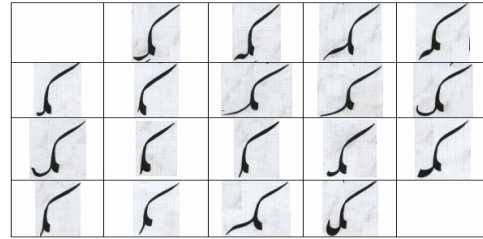


Figure 52: Medial Shapes of Kaf [44]

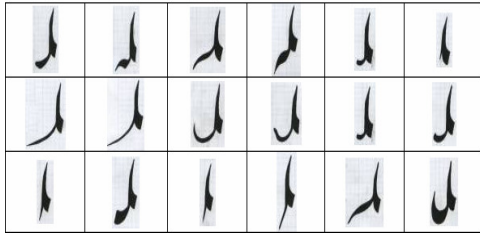


Figure 53: Medial Shapes of Laam [44]

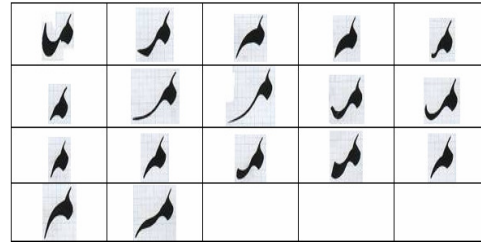


Figure 54: Medial Shapes of Meem [44]

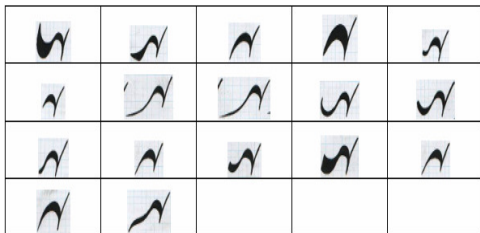


Figure 55: Medial Shapes of Goal Hay [44]





Figure 56: Medial Shapes of Do chasmi Hay [44]

FINAL SHAPES

Following are the final shapes of some of the characters.

Table 6 : Final Shapes of Nastaleeq Characters

| Character shapes | Characters |
|---|---|
|  |  |

| | |
|--|--|
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

- It is written from top right to bottom left with each ligature titled at approximately 45 degree. That is to say that it is written diagonally. It is space- conserving style of writing and takes approximately 40% less space than Naskh [44].

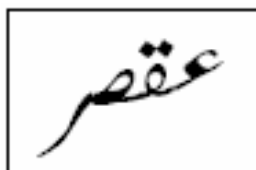


Figure 57 : Nastaleeq Diagonality

- It has fine or fragile joins. The strokes are thin where a character shape joins with another character. The character shapes themselves, are generally thick as compared to the joins. Thus, there is an alternating sequence of thick and thin strokes in the ligatures [44].

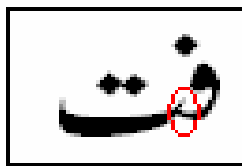


Figure 58 : The Join where Fay Connects to Tay is Thin as Indicated by the Red Circle

- The joins are formed by cusp-like shapes, which are concave upwards and have their initial end higher than the final end [44].

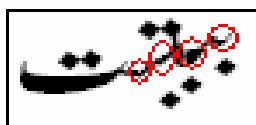


Figure 59 : Cusp-liked joins are circled in red

- It has complex Mark placement rules e.g. for Nukta' s and diacritics [44].



Figure 60 : Two Different Placements of Dots of Chay

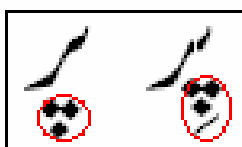


Figure 61 : Two Different Placements of Dots of Pay due to Zair

- There is not a fixed baseline on which characters are written. The baseline changes with the addition of each character. That is to say that there are multiple baselines as it is different for all characters within a ligature [45].

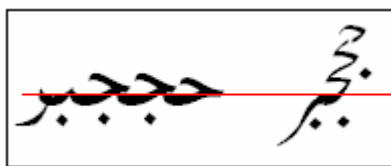


Figure 62 : Naskh Versus Nastaleeq Writing Style

The exit point of one character is joined with the entry point of next character. In order to do this the shape of the character is changed.



Figure 63 : Different Base Line for Different Characters

- The overlapping problem is present not only in characters but in ligatures as well, as the calligrapher always tries to produce a piece of script, which is beautiful and catches the eye [44].



Figure 64 : Overlap Between the Ligatures

- There is no monotony among shapes of letter at same position i.e. initial, final and medial [44].



Figure 65 : Different Shapes of Kaf at the Same Medial Position

3.4 Noori Nastaleeq

In 1970, Pakistani calligrapher Ahmed Mirza Jamil revolutionized Urdu calligraphy by inventing a new technique of joining Urdu letters and Words called Noori Nastaleeq. Thus made, the use of computer for Nastaliq styles a reality [39].

Mr Ahmed Mirza Jamil calligraphically designed each ligature so as to make it suitable for feeding it into the memory of a Computer. The resulting script was difficult, as it was not only based on the pre-defined formulas but also on the aesthetic sense of the calligrapher. Computerized composing replaced manual calligraphy but retained the elegance and beauty of the written language with Nastaliq Script. Today the fast-computerized calligraphy has been admired

all over the world and countless dailies; magazines, journals and books are being published in almost no time [39].

In 1982, Noori Nastaliq was designated as an Invention Of National Importance and Ahmed Mirza Jamil was awarded Tamgha - I - Imtiaz by the Government of Pakistan [39].

4 Problem Statement

Significant amount of data that is available on the Internet is in the English language. Electronically available Urdu data is mostly in image form, which is very difficult to process. Printed Urdu data is the root cause of the problem. So for the rapid progress of Urdu language we need OCR systems, which can help us to make Urdu data available for the common person.

Majority of the OCR systems available today are for the printed Arabic Script with reasonable level of accuracy. Little work has been done on Urdu language that is mainly towards the recognition of Naskh script. It is a simpler script than Noori Nastaleeq [51].

A widely used script in newspapers, governmental documents and books is the Noori Nastaleeq script but there is a very little reported effort at developing OCR systems for Noori Nastaleeq script.

Urdu, the official language of Pakistan, is mostly used in newspapers and in little government documentation. 80 % of the population uses their native languages for communication. But still no one has done a significant work for the development of the OCR system.

Therefore, the lack of interest of people and unavailability of fonts and Unicode for a long time proved to be big hurdles in the development and advancement of Urdu OCR system [47].

All the above facts turned to be the motivation behind our work. Moreover no one has used HMM technique for the recognition of “Noori Nastaleeq Script”. Emphasizes of this project is to convert a scanned textual image into an editable file format with the reasonable level of accuracy.

The development of this project will cover another milestone towards the advancement of Urdu language as well as help us to maintain computerized Urdu data.

4.1 Problem Scope

The scope of this work is based on the following assumptions:

- Noori Nastaleeq developed by Mirza Ahmed Jamil will be used for analysis, development and testing.
- The system will be optimized for a single font-size i.e. point size 36.
- All the shapes (isolated, initial, final and medial) of the following class will be handled.

Table 7 : Shapes that will be Recognized

| Characters | Class | Sr. no |
|------------|-------|--------|
| آ ا | ا | 1 |
| ب پ ت ث ط | ب | 2 |
| و ڈ ذ | و | 3 |
| ص ض | ص | 4 |
| ع غ | ع | 5 |
| ی | ی | 6 |

- Only the single columned document without formatting is handled. As shown below

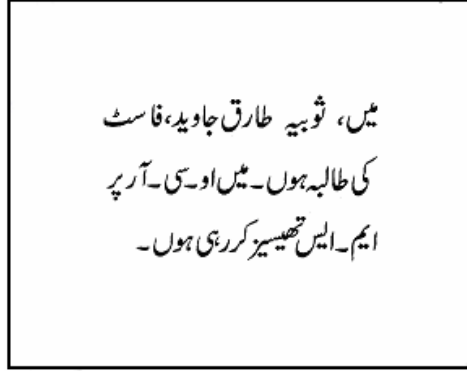


Figure 66 : Single Columned Document without Formatting

- Only the Urdu Noori Nastaleeq text without Punctuation marks (^ " ! : " ' ? () [] { } .), Arithmetic operators and symbols, Urdu numerals (* ۰ ۱ ۲ ۳ ۴ ۵ ۶ ۷ ۸ ۹), diacritics, signs, complex ligatures, etc is handled.
- The text in the table will not be handled.
- The system will be able to process only already pre-processed images. As we will be concentrating on the main problem here i.e. recognition of Noori Nastaleeq and will ignore these issues as other researchers have already efficiently addressed them.
- No post processing will be done.

5 Methodology

From the literature review we conclude that the statistical classifiers are better pattern matching technique. We have seen two statistical classifiers commonly used for character recognition .i.e. Neural Network and Hidden Markov Model (HMM).

Table 8 : Comparison of Neural Network and HMM

| Sr. | Neural Network | Hidden Markov Model |
|-----|---|--|
| 1 | Remarkable contribution to the development of OCR system | Remarkable contribution to the development of OCR system in case of hand writing recognition. |
| 2 | Training of neural networks is a difficult task as the number of samples should be same for each trainable data and the network needs to be built again | On the other hand training of HMM is very straight forward. Network is only updated with the addition of each new data, not rebuilt. Number of samples for different |

| | | |
|---|---|--|
| | with the addition of each new data. So training is a time consuming process. | data can vary. |
| 3 | Addition of training data gives the boost to the performance but makes the network very complicated. Mesh kind of network is created which increases the computational effort and time to train the data. | The addition of training data increases the recognition accuracy but there is no effect on training process. |
| 4 | High recognition rates with a small set of data. But the performance declines with the increase of data set. The system fails with the very large set of data. | High recognition rates with all kind of data. |

From the above table we can conclude that HMM is a better pattern matching technique which has a high accuracy rate even with large data set.

So our system is based on HMM. The recognition process is divided into two steps

- Training
- Recognition

Methodology of training and recognition is as shown in the diagrams given below.

Flow Chart For Training

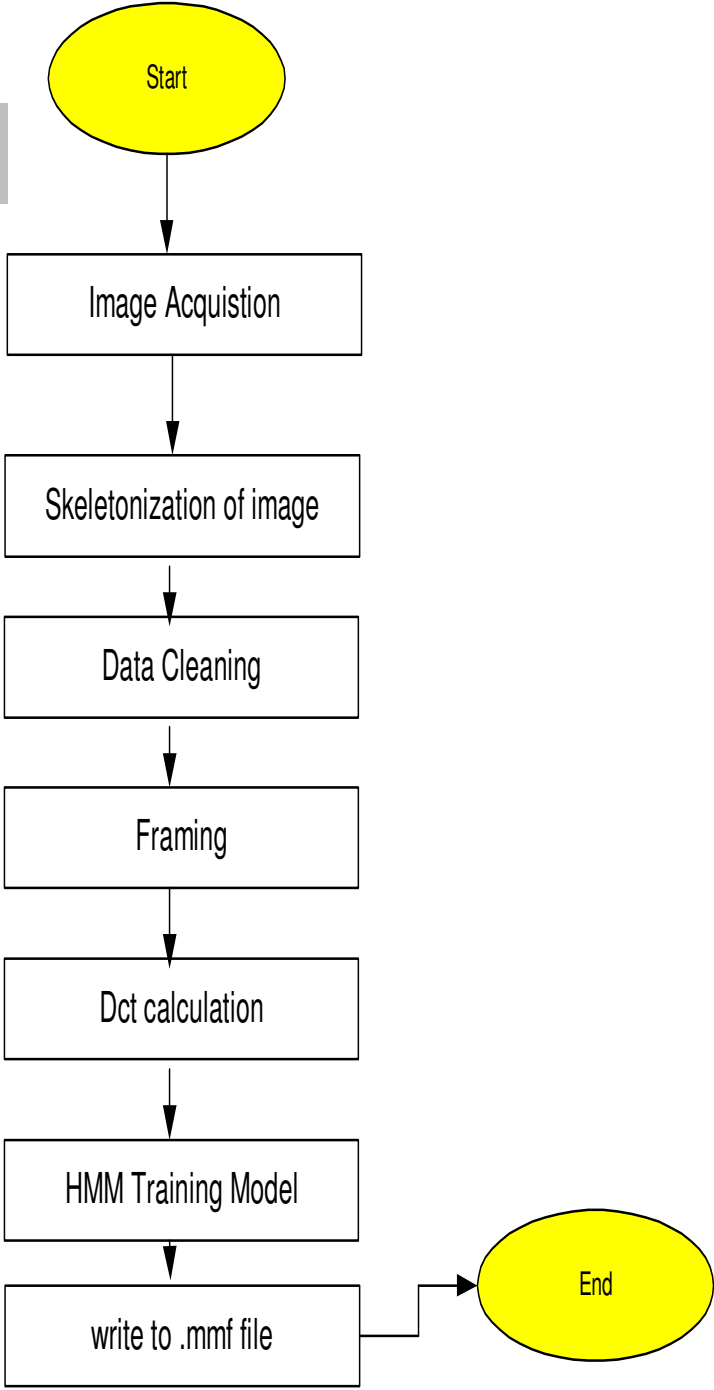


Figure 67 : Flow Chart for Training

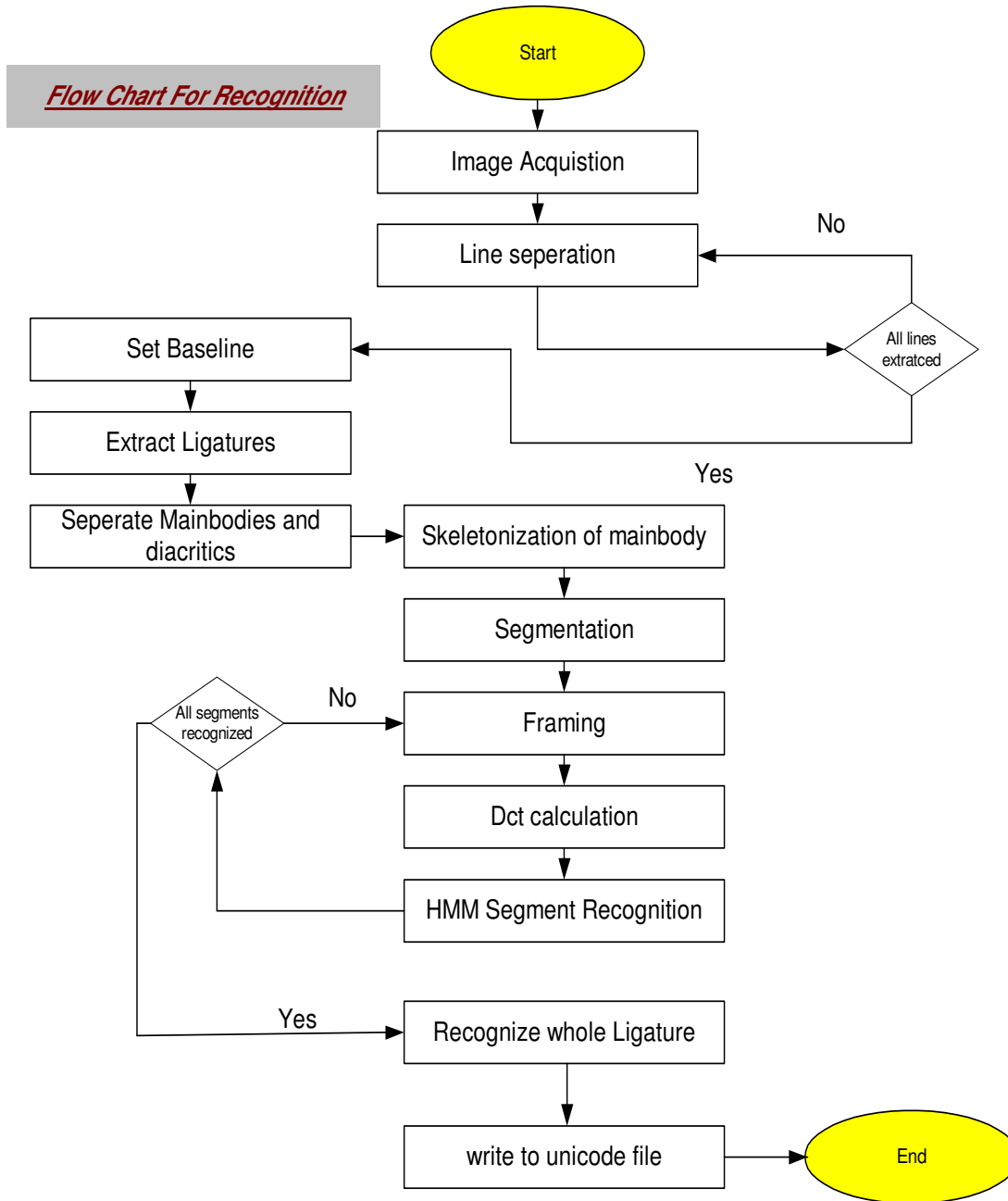


Figure 68 : Flow Chart for Recognition

5.1 Scan Image / Open Image

The image, which needs to be converted, has to be obtained in this very first phase of OCR systems. The OCR system needs a scanned image containing Urdu text as input. Since the

preprocessing will not be done so it is assumed that the image will be kept properly for scanning to avoid skew in the image and there will be no noise and distortion.

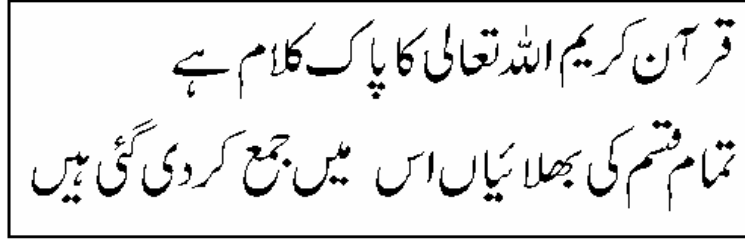


Figure 69 : Scanned Image

5.2 Separate Line of Text

Our proposed system automatically detects and separates individual text lines from the image using horizontal and vertical projection of pixels.

A projection profile is a histogram giving the accumulated sum of ON pixels along rows. It is in fact, a one-dimensional array where each element denotes the number of ON pixels along the respective row in the image. The trough between two consecutive peaks in the horizontal profile will denote the boundary between two text lines. Similarly through vertical projection of each line the right and left of line is marked.

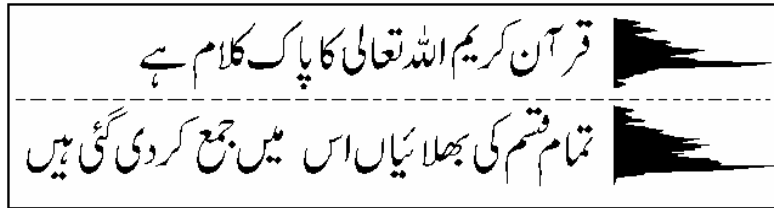


Figure 70 : Separate Lines from the Text by Using Horizontal Histogram

After the lines separation, the boundaries of the lines are defined using vertical histogram.

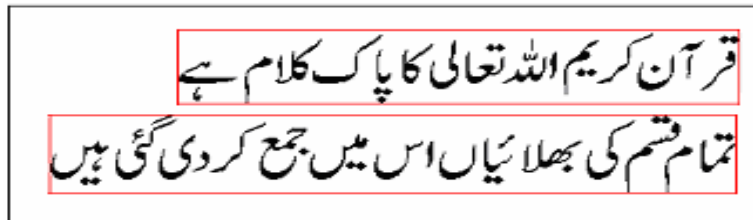


Figure 71 : Boundary of Lines

5.3 Set Base Line

Since Urdu is written on a horizontal line called baseline so we need to mark it to differentiate between main bodies and diacritics. The information of horizontal projection of pixels is used to set the baseline of a line. Row that contains the maximum number of pixels will be set as a baseline.

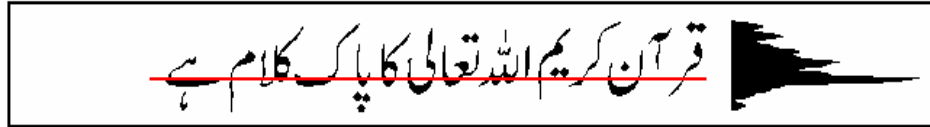


Figure 72 : Base Line Detection by Using Horizontal Histogram

One problem that arose when using this method is that sometimes baseline may be set towards the top of line. To solve this problem we set the baseline by selecting the row containing maximum number of pixels from lower half of line. Moreover instead of using a single row as a base line we used band of baseline consisting of 5 to 10 lines of row. This base line is used to distinguish between main body and diacritics.

5.4 Isolate Ligature and Diacritics

One line will be taken at a time and then the ligatures and diacritics are isolated from it. The connected components that lie on the baseline will be considered as main bodies of ligature and rest are considered as diacritics.

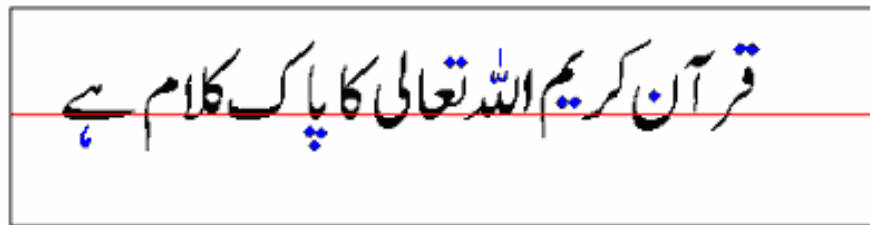


Figure 73 : Blue are the Diacritics while Black are the Ligatures

5.5 Segmentation of Main body

The proposed OCR system applies the Skeletonization process on the main bodies, to extract a region-based shape features representing the general form of an object.

“Skeletonization is a process for reducing foreground regions in a binary image to a skeletal remnant that largely preserves the extent and connectivity of the original region while throwing away most of the original foreground pixels” [46] .

The definition of the skeleton will be illustrated by the prairie-fire analogy: the fire is set on the boundary of an object and the loci where the fire fronts meet and quench each other is the skeleton [46].

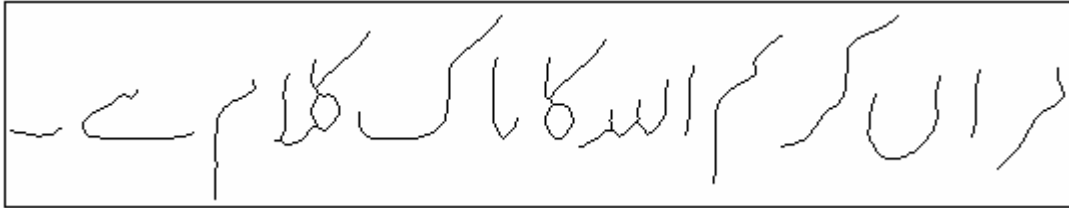


Figure 74 : Sketetonized Image

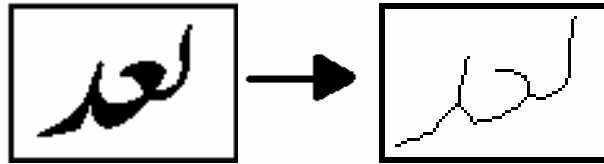


Figure 75 : Sketetonized Image of the Word Baad

As we know that Urdu is written right to left, so that means the starting point of the ligature (the first pixel of the first character in the ligature) will lie some where around the right side of the ligature. At this point we might think that the right most one neighbor pixel will be the starting point.

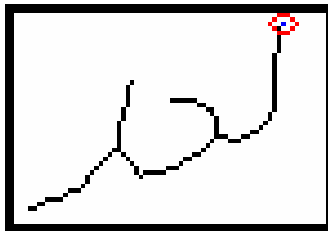


Figure 76 : Right Most Encircled Pixel is the Starting Point of the Ligature

Though this concept can be valid for the Naskh script but the overlapping nature of Nastaleeq script makes it invalid. Because in certain cases, the rightmost one-neighbor pixel is not necessarily the starting point of the ligature.

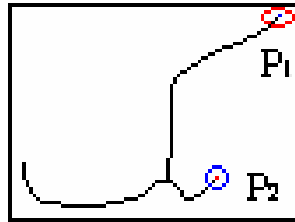


Figure 77 : The Rightmost Pixel P1 is not the Starting Point of the Ligature whereas the Pixel P2 is the Actual Starting Point of the Ligature

In the above example though the pixel P1 is the rightmost pixel but that is not the starting point of the ligature. Same is the case in the example below.

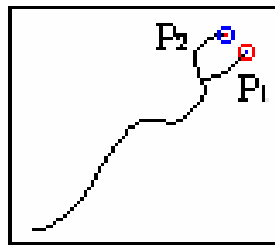


Figure 78 : The Rightmost Pixel P1 is not the Starting Point of the Ligature whereas the Pixel P2 is the Actual Starting Point of the Ligature

Computer is not so much intelligent as human being so it cannot judge what will be the starting point of the ligature in Noori Nastaleeq Script. Since Nastaleeq is written diagonally and characters overlap their preceding characters so that means there will be one and only one ending point of the ligature and there is no ambiguity in that since no character overlaps the last character of the ligature. So for this reason we have tried to find the last point in the ligature, which is written when writing with pen.

The skeleton is traversed from left to right i.e. the left most ending point of the last letter will be taken as the initial node.

In order to find the starting point for the traversal, first the lowest-left most pixel P1 having only one neighbor is found and declared as potential starting point.

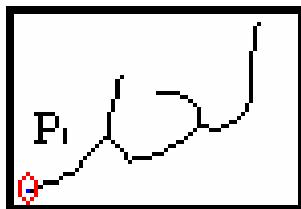


Figure 79 : The Starting Point of the Ligature is Circled with Red

Then left most pixel of the ligature is checked for vertical connected line to determine if the last character in the ligature is Alif. If the last character is Alif then highest-left most pixel P2 is declared as the starting point otherwise P1 is considered the starting point.

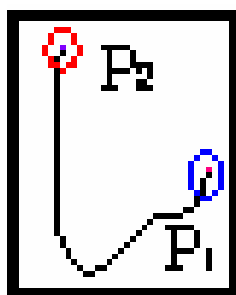


Figure 80 : The Actual Starting Point P2 of the Ligature is Circled with Red while the Potential Starting Point P1 is Circled with Blue

The order of directions for the traversal is west, north, south and east. On the way of traversal any pixel having more than two black neighboring pixels will be considered as break point. The body will be segmented at the break points and all the edges will be disconnected from this point.

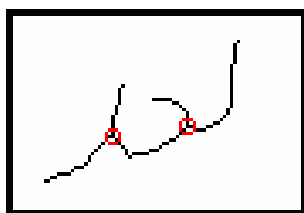


Figure 81 : Break points in a Sketetonized Image Circled with Red

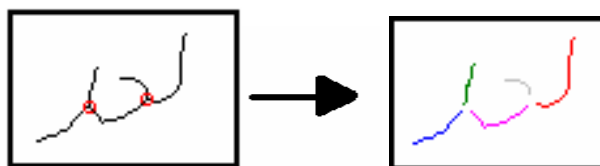


Figure 82 : Segments in the Ligature are Colored Differently

The segmentation may result in over segmentation. As in the above case the letter ain is over segmented into two segments.

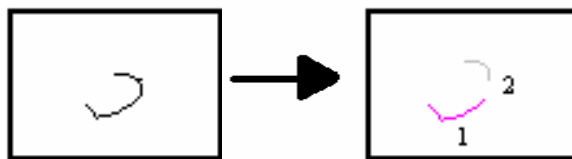


Figure 83 : Over Segmentation in Medial Ain

Similarly it may result in under segmentation if there is no proper junction point between the two letters.

For example in the word bib there is no break point and whole ligature is considered as one segment.

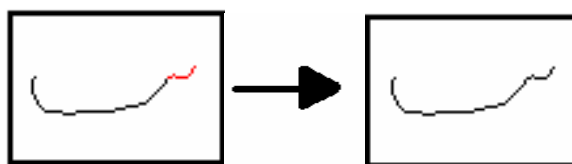
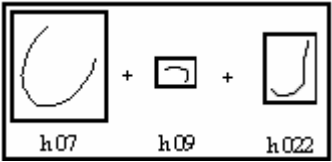

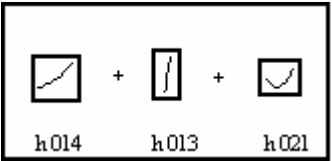
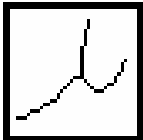

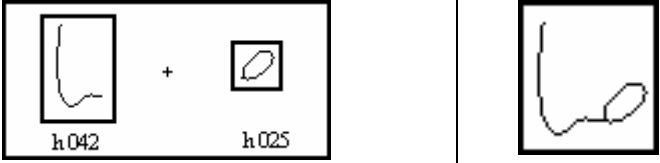
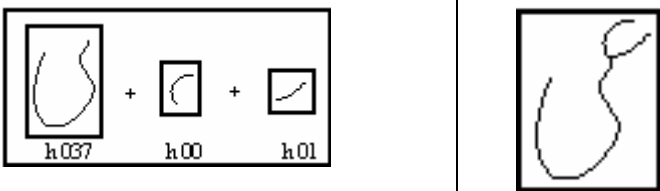


Figure 84 : Under Segmentation in the Word Bib

The following table gives the segmentation in few ligatures.

Table 9 : Segmentation of the Ligatures

| Sr. | Segments | Ligature |
|-----|---|---|
| 1 |  |  |
| 2 |  |  |

| | |
|---|--|
| 3 |  |
| 4 |  |
| 5 |  |

5.6 Framing

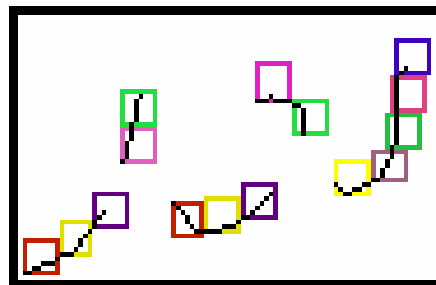


Figure 85 : The Framing of Segmented Word Baad

After skeletonization, the frames of each segment are fed to HMM for training and recognition. We took a frame of 8x8 size, which constitutes of only 8 black pixels of the segment. The frame traversal is same as ligature traversal explained above. We experimentally found out that larger the frame size more ambiguous would be the system. We tested the system with 5x5, 8x8, 9x9, 12x12 and 16x16 frame sizes and found out that 8x8 gave us the best results.

5.7 Hidden Markov Model

Our selected research technique is Hidden Markov Model (HMM), which has an ability to perform recognition with great ease and efficiency. Language-independent training and

recognition methodology; automatic training on non-segmented data; and simultaneous segmentation and recognition are the useful aspects of using HMM in developing continuous recognition technology [51].

5.7.1 HMM Training

When pre-processing is complete, before starting training process, the HMM parameters must be properly initialized with training data in order to allow a fast and precise convergence of the training algorithm.

Each segment is considered as a separate HMM. There are total of sixty HMMs.

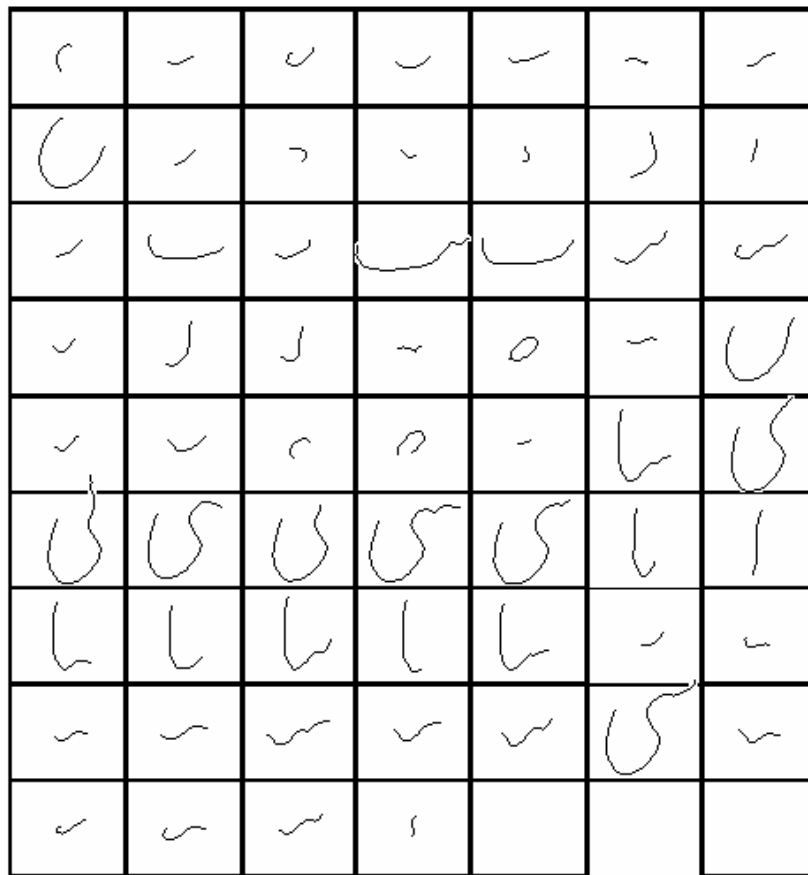


Figure 86 : HMM Model Used

In order to cater all sorts of noise and variations in the image we need to collect at least 100 samples of each hmm model. [See appendix A]

For each segment we have to define the number of states required [see appendix B].

5.7.2 HMM Recognition

Before recognizing, we follow these steps:

- Build the task Grammar
- Build the Dictionary
- Make Network using HMM command

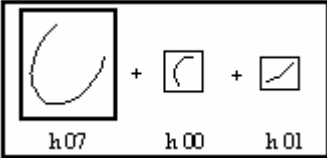

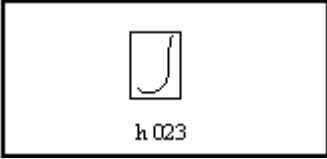

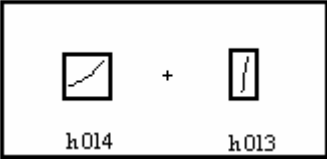

After training the model on the given data, the recognition process is performed. In recognition process the skeletonized ligature is first segmented and then each segment is fed to HMM for the segment recognition. HMM gives the output, which has the maximum probability given the observation sequence.

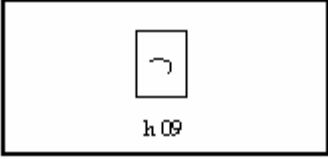

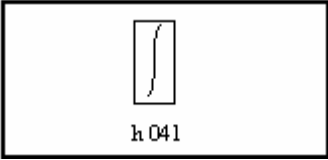

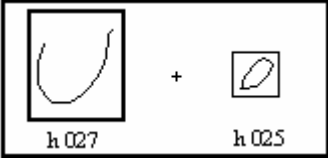
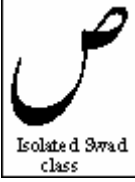
5.8 Recognizing the Ligature

Once the segments are recognized, the rules are applied to recognize the ligature.

The following table gives us the examples of some of the rules.

Table 10: Rules

| Sr. | Segments | Letter |
|-----|---|---|
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |

| | | |
|---|--|---|
| 4 |  <p style="text-align: center;">h 09</p> |  <p style="text-align: center;">Medial Ayn class</p> |
| 5 |  <p style="text-align: center;">h 041</p> |  <p style="text-align: center;">Medial Alif class</p> |
| 6 |  <p style="text-align: center;">h 027 + h 025</p> |  <p style="text-align: center;">Isolated Waw class</p> |

Let's take an example of the word baad.

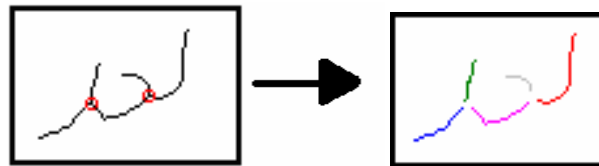







Figure 87 : Segmentation in the Word Baad

Table 11: HMM Names

| Sr. | HMM | HMM name |
|-----|---|----------|
| 1 |  | h022 |
| 2 |  | h09 |
| 3 |  | h029 |
| 4 |  | h013 |
| 5 |  | h014 |

Rules:

Rule 1: Bay → h022

Rule 2: Ain → h09 + h029

Rule 3: Dal → h013 + h014

So HMM will give us the HMM names in the following order

| | | | | | |
|-----------------|------|------|------|-----|------|
| HMM name | h014 | h013 | h029 | h09 | h022 |
| Location | 1 | 2 | 3 | 4 | 5 |

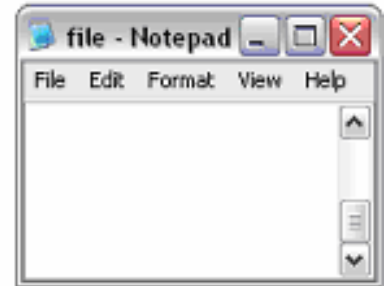


Figure 88 : The Results Returned by HMM and Empty Unicode File

Suppose we have a pointer to point the current HMM name. Now starting from the right most HMM, our pointer will be at location 5. According to the rule 1, h022 will be converted to bay. So we will write bay in the Unicode file.

| | | | | | |
|-----------------|------|------|------|-----|------|
| HMM name | h014 | h013 | h029 | h09 | h022 |
| Location | 1 | 2 | 3 | 4 | 5 |



Figure 89 : The Pointer is at Location 5. After Applying Rule 1, Bay is Written in Unicode File

Now our pointer will be at location 4 and according to rule 2, ain is recognized and written in the file, which results in a ligature comprising of bay-ain.

| | | | | | |
|-----------------|------|------|------|-----|------|
| HMM name | h014 | h013 | h029 | h09 | h022 |
| Location | 1 | 2 | 3 | 4 | 5 |



Figure 90 : The Pointer is at Location 4. After Applying Rule 2, Bay-Ain is Written in Unicode File

Our pointer now jumps to the location 2. According to rule 3, h013 and h014 will form dal and is written in the Unicode file.

| | | | | | |
|-----------------|------|------|------|-----|------|
| HMM name | h014 | h013 | h029 | h09 | h022 |
| Location | 1 | 2 | 3 | 4 | 5 |

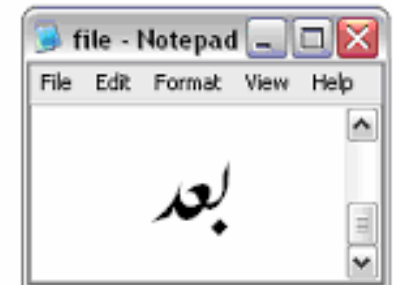
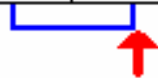


Figure 91 : The Pointer is at Location 2. After Applying Rule 3, Bay-Ain-Dal is Written in Unicode File

So we get the ligature baad, after applying the rules.

6 Results

This Section gives the results of the work done.

6.1 Purpose

The purpose of scientifically evaluating and analyzing Urdu Nastaleeq OCR is to measure its accuracy in recognizing ligatures.

6.2 Sample

For analyzing the OCR the words were extracted from Nokia dictionary, which constituted the letters from the above-mentioned classes. Three or more samples of each word were taken. The Urdu words were written in font Noori Nastaleeq and font size 36. The pages were scanned at dpi 150.

6.3 Method

We tested each type of letter without diacritics by recognizing it using Urdu Nastaleeq OCR. Three or more tokens of each word were used. Accuracy is calculated based on recognition. Following measures were used to check system accuracy.

- Percentage of recognition of tokens
- Error rate of tokens

6.4 Test Results

The accuracy is calculated on letters and ligatures basis.

6.4.1 Character level Results

Accuracy of each letter is calculated separately. It has been tried that all the forms (initial, medial, final and isolated) of the letter are covered in the testing data.

6.4.1.1 Class of Alif

Table 12: Accuracy of Alif

| | |
|---------------------------------|-------|
| Number of tokens | 765 |
| Number of tokens recognized | 761 |
| Percent tokens recognized | 99.5% |
| Error rate in token recognition | 0.5 |

6.4.1.2 Class of Swad

Table 13: Accuracy of Swad

| | |
|-----------------------------|-----|
| Number of tokens | 150 |
| Number of tokens recognized | 120 |
| Percent tokens recognized | 80% |

| | |
|---------------------------------|-----|
| Error rate in token recognition | 20% |
|---------------------------------|-----|

6.4.1.3 Class of Dal

Table 14: Accuracy of Dal

| | |
|---------------------------------|-------|
| Number of tokens | 459 |
| Number of tokens recognized | 456 |
| Percent tokens recognized | 99.3% |
| Error rate in token recognition | 0.7% |

6.4.1.4 Class of Choti Yeh

Table 15: Accuracy of Choti Yeh

| | |
|---------------------------------|-------|
| Number of tokens | 132 |
| Number of tokens recognized | 117 |
| Percent tokens recognized | 88.6% |
| Error rate in token recognition | 11.4% |

6.4.1.5 Class of Ain

Table 16: Accuracy of Ain

| | |
|---------------------------------|-------|
| Number of tokens | 339 |
| Number of tokens recognized | 321 |
| Percent tokens recognized | 94.7% |
| Error rate in token recognition | 5.3% |

6.4.1.6 Class of Bay

Table 17: Accuracy of Bay

| | |
|---------------------------------|-------|
| Number of tokens | 1053 |
| Number of tokens recognized | 1000 |
| Percent tokens recognized | 94.9% |
| Error rate in token recognition | 5.1% |

6.4.1.7 Cumulative Results

Table 18: Cumulative Results

| | |
|------------------|------|
| Number of tokens | 2898 |
|------------------|------|

| | |
|---------------------------------|--------|
| Number of tokens recognized | 2775 |
| Percent tokens recognized | 95.76% |
| Error rate in token recognition | 4.24% |

6.4.2 Ligature Level Results

It has been tried that maximum number of ligature be covered in the training data. Again three repetitions of each ligature are taken.

Table 19: Ligature Level Accuracy

| | |
|---------------------------------|-------|
| Number of tokens | 1692 |
| Number of tokens recognized | 1569 |
| Percent tokens recognized | 92.7% |
| Error rate in token recognition | 7.3% |

6.5 Examples

Following figures show some results of OCR.

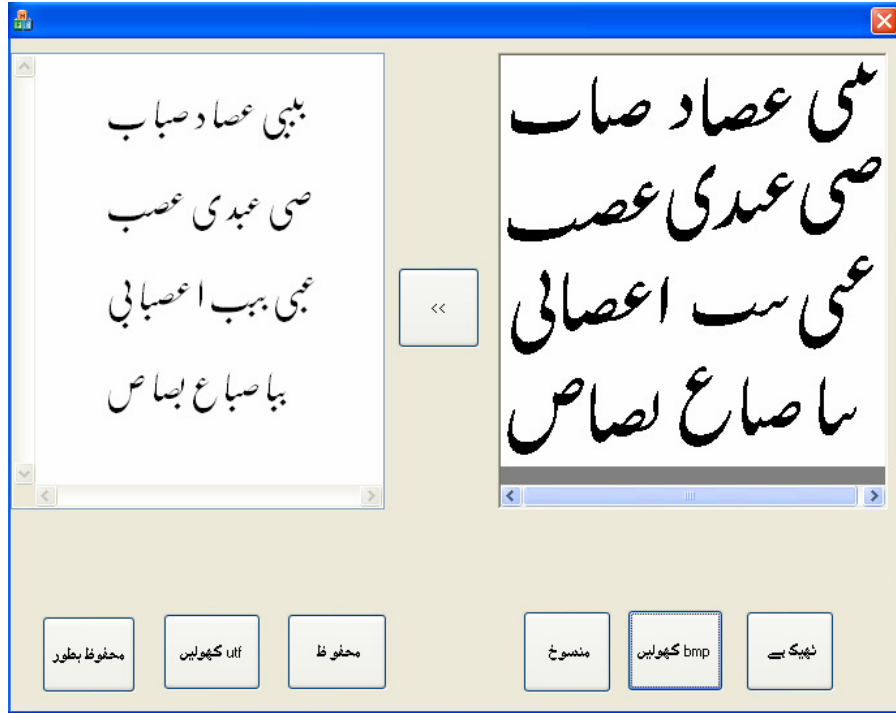


Figure 92 : Output 100% Accuracy

The result obtained in the above figure is 100 % correct. The .bmp file is on the right side of the figure and the result is on the left side.

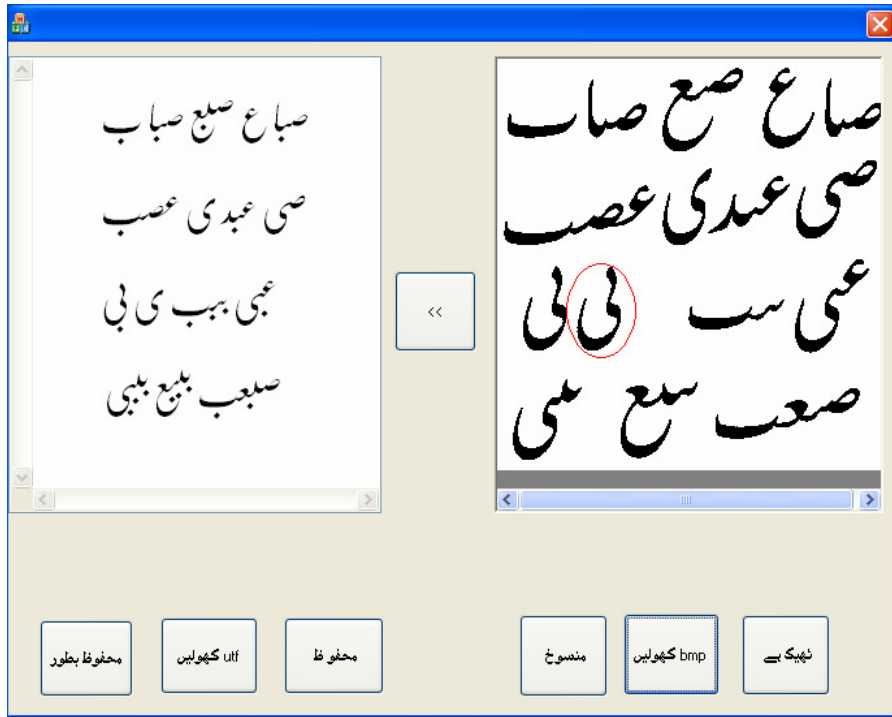


Figure 93 : Output with Some Error Due to Noise and Distortion

The result in the above figure is not 100% correct. The Word “صبا” is not recognized properly because of the noise and distortion in the image.

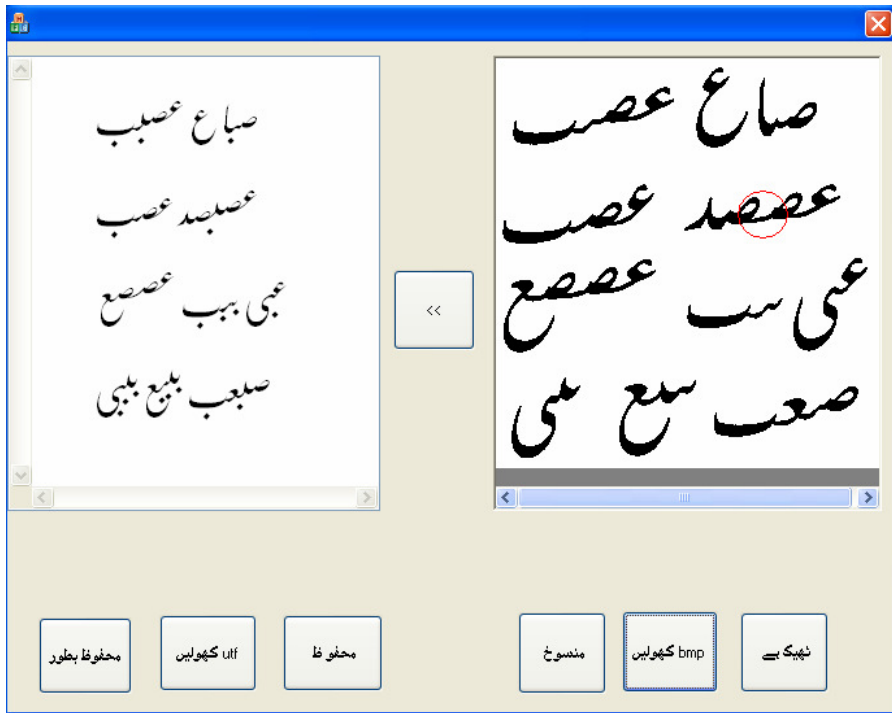


Figure 94 : Output with Training Error

The result in the above figure is not 100% correct. The Word “عصص” is not recognized properly because of the training error.

7 Discussion

The main focus of the project was to do research and make a ligature in dependent OCR capable of recognizing Urdu Nastaleeq font using HMM technique. This main objective of our project has been accomplished. Some letters were not recognized correctly. Following were the problems due to which OCR was not able to recognize these letters.

7.1 Distortion in the Image

The distortion or the noise in the image can also affect the output of the recognizer.

7.1.1 Problem Description

Sometimes due to noise or poor scanning the actual shape of the letter is distorted. As discussed above a small change in the original image will greatly affect its skeleton. In this case, the letter is not recognized correctly.

7.1.2 Example

During scanning the “بی” is distorted. Total number of windows and DCT values differ to great extent. This results in erroneous recognition of letters.

For example we have a word bibi in the figure 89. First bay-yeh is not recognized properly where as in the second case it is recognized correctly.

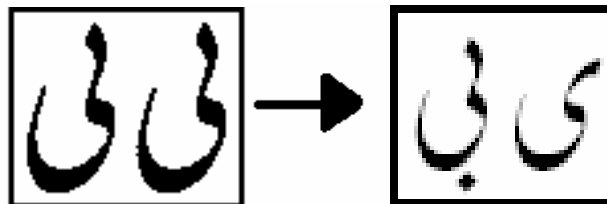


Figure 95 :Bay-Yeh not Recognized Properly due to Training Error

The reason is that there is some distortion in the first bay-yeh which changed its skeleton greatly.

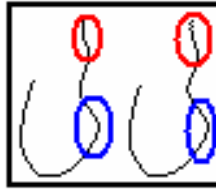


Figure 96 : Encircled are the Differences in the Skeleton of the Ligatures

7.1.3 Solution

We have trained OCR on 100 samples (at least) of each segment. This problem may be removed by training the HMM on more samples of each segment and giving original segment as HMM input instead of giving skeletonized segment. This will provide more information to HMM for the final recognition. Moreover if the pre-processing is applied before the recognition and training step, the results will be improved significantly.

7.2 Similarity in Shape

The similarity in the shapes of different characters can lead the recognizer to confusion.

7.2.1 Problem Description

Sometimes the shapes of different letters in different forms are so similar that they give approximately same state transition probabilities. So when these inputs are given to HMM for recognition, same state sequence of HMM is obtained. Erroneous output is thus obtained due to the slight similarity in shape. This problem cannot be solved by changing number of states during training.

7.2.2 Examples



Figure 97 : Similarity in Shapes of Swad-Bay-Dal and Swad-Dal

The shape of the letters bay and last stroke in swad are similar to each other when written in Noori Nastaleeq font. Sometimes the OCR is not able to recognize segments like these correctly because of the similarity in shape.



Figure 98 : Similarity in Shapes of Bay and Last Stokes of Ain and Swad

Another such problem can be seen in “تتبع” and “عصص” shown above. The shape of bay is more similar than different to the last stokes of swad and ain. Sometimes they are recognized correctly and sometimes they are not.

7.2.3 Solution

We used a sliding non-overlapping window of size 8*8 to calculate DCT. This problem can be removed by making the sliding window over-lapping. By using overlapping windows finer details of segment shape can be also be covered. Moreover many such problems will be automatically removed when diacritics are handled. For example sometimes swad-dal is misrecognised as swad-bay-dal due to the similarity in shapes as shown in the figure above. When no diacritic will be found for bay, we will come to know that the portion of swad is wrongly classified as bay.

7.3 Inconsistency in Font

Inconsistent font can be the major cause of low accuracy rate.

7.3.1 Problem Description

The font Noori Nastaleeq shows very inconsistent behavior i.e. two words, with the only difference of diacritic placement, have different shapes. Since the font is ligature dependent and is hand written so there is so much dissimilarity between the shapes of letters belonging to the same class.

7.3.2 Example

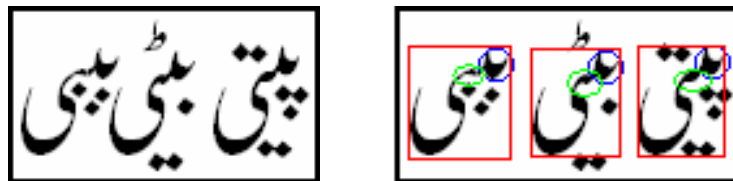


Figure 99 : Dissimilarity in Shapes of Same Ligature with Different Diacritic Placement

Since there is so much inconsistency so it might happen that some form of the shape is missed in data collection step and that results in an error in recognition step. So much diversity in the same class leads the HMM to confusion.

Another such example is given below.



Figure 100 : Another example of the dissimilarity in shapes of same ligature with different diacritic placement

7.3.3 Solution

We have trained OCR on 100 samples (at least) of each segment [see appendix A]. This problem may be removed by training the HMM on more samples of each segment and giving original segment as HMM input instead of giving skeletonized segment. This will provide more information to HMM for the final recognition. Moreover if the pre-processing and post-processing steps are applied, the results will be improved significantly.

8 Conclusion

In this chapter we have discussed the results and accuracy of OCR. The individual letter accuracy of OCR in token recognition is 95.76% and in type recognition it is 100%. The ligature accuracy of OCR in token recognition is 92.73% and in type recognition it is 91.43%. This accuracy will further improve when diacritics will be handled and pre-processing is applied. The accuracy is calculated by testing the word from the Nokia dictionary. By seeing the results, we can say that HMM technique is good for recognizing letters. The results can be more accurate if sliding window approach is used on the original segment without thinning and pre and post processing is done.

9 Future Work and Enhancement

Many enhancements can be made in Urdu Nastaleeq OCR.

- The software can be made font independent
- It can be made font size independent
- Automatic noise removal
- Automatic skew detection and correction
- Diacritics can be handled
- Due to time limitation this software is developed for the above mentioned 6 classes only; other letters can be catered afterwards.
- The performance can be further increased by having original stokes as HMM input instead of thinned stokes.
- A deep analysis of Noori Nastaleeq font can be done for further usage in rules generation of OCR.
- Post processing can be done.

The Urdu OCR system is an area, which requires much more research to make it efficient and suitable for consumer use. This thesis is a model towards implementing a more efficient system and still has a lot to be desired.

Reference

- [1]. A. J. Elms and J. Illingworth, "***Modeling polyfont printed characters with HMMs and a shift invariant Hamming distance***", Proceedings of the Third International Conference on Document Analysis and Recognition (Vol. 1), pg.504, August 14-15, 1995.
- [2]. M. Mohamed and P. Gader, "***Handwritten word recognition using segmentation-free hidden Markov modeling and segmentation-based dynamic programming techniques***", IEEE Trans. Pattern Anal. Mach. Intel. Vol.18, no.5, pg.548-554, 1996.
- [3]. A. Kornai, "***Experimental HMM-Based Postal OCR System***", Proceeding of International Conference. Acoustics, Speech, Signal Processing, vol. 4, pg. 3,177-3,180,Munich, Germany, 1997.
- [4]. M. S. Khorsheed and W.F. Clocksin, "***Structural Features Of Cursive Arabic Script***", in Proceeding of British Machine Vision Conference, pg.1285-1294, 1999.
- [5]. Zhidong Lu, Issam Bazzi, Andras Kornai and John Makhoul, "***A Robust, Language-Independent OCR***", System. In the 27th AIPR Workshop: Advances in Computer Assisted Recognition, SPIE, 1999.
- [6]. Marija Bojovic & Milan D. Savic, "***Training of Hidden Markov Models for Cursive Handwritten Word Recognition***", in 15th International Conference on Pattern Recognition (ICPR'00) - Vol1 pg. 1973, September 2000.
- [7]. M. Pechwitz and V. Maergner, "***HMM based approach for handwritten Arabic word recognition using the IFN/ENITdatabase***", Proceeding of 7th ICDAR 2003, pg. 890-89, 2003.
- [8]. M.S Khorsheed, "***Recognizing Handwritten Arabic manuscripts using a single hidden Markov model***", Pattern Recognition Letters, Vol.24, pg.2235-2242, 2003.
- [9]. E. Lecolinet, "***Cursive script recognition by backward matching***", Proceeding of Sixth International Conf. Handwriting and Drawing, pg. 89-91, 1993.
- [10]. Juan-Carlos Perez, Enrique Vidal and Lourdes Sanchez, "***Simple and effective feature extraction for optical character recognition***", the 5th Spanish Symposium on Pattern recognition and images analysis, 1994.

- [11]. Badr Al-Badr and Robert M. Haralick, "*Segmentation Free Word recognition with application to Arabic*", in Proceeding of International Conference on Document Analysis and Recognition, pg.355-359, 1995.
- [12]. D. Megherbi, S.M. Lodhi and J.A. Boulenouar, "*A fuzzy logic-based technique for Urdu Character Representation and Recognition*", Proceedings of SPIE -- Vol 3962, pg. 13-24, April 2000.
- [13]. Sofien Touj, Najoua Essoukri Ben Amara and Hamid Amiri, "*Generalized Hough Transform for Arabic Optical Character Recognition*", Document Analysis and Recognition, Proceedings of Seventh International Conference, Aug. 2003.
- [14]. Stephen Pearce, Maher Ahmed, "*An Evolutionary Algorithm for General Symbol Segmentation*", ICDAR 2003:pg 726-730,2003.
- [15]. Simon Günter and Horst Bunke, "*A New Combination Scheme for HMM-Based Classifiers and its Application to Handwriting Recognition*", ICPR (2) 2002:pg 332-337, 2002.
- [16]. Somaya Alma'adeed, Colin Higgins, and Dave Elliman, "*Recognition of Off-Line Handwritten Arabic Words Using Hidden Markov Model Approach*", 16th International Conference on Pattern Recognition (ICPR'02) - Vol 3, pg. 30481, August 2002.
- [17]. H. Bunke, M. Roth and E. C. Schukat-Talamazzini, "*Off-line cursive handwriting recognition using hidden Markov models*", Poll. Recogn. 28, 9 (1995): pg. 1399-1413. 1995.
- [18]. R. El-Hajj, L. Likforman-Sulem and C. Mokbel, "*Arabic Handwriting Recognition Using Baseline Dependant Features and Hidden Markov Modeling*", in the 8th International Conference on Document Analysis and Recognition, ICDAR 2005, Seoul, Korea, (2005).
- [19]. C. Bose and S. Kuo, "*Connected and degraded text recognition using hidden Markov model*", in Proceeding of 11th International Conference Pattern Recognition, pg 116-119,1992.
- [20]. Elms, A.J., "*A Connected Character Recognizer Using Level Building of HMMS*", In: Proceeding of 12th International Conference Pattern. Recognition, pg 439-442, October 1994.

- [21]. Amlan Kundu, Yang He, Mou-Yen Chen, "*Alternatives to Variable Duration HMM in Handwriting Recognition*", PAMI(20), Vol. 20, No. 11, pg. 1275-1280, 1998.
- [22]. Zahra Shah and Farah Saleem, "*Ligature Based Optical Character Recognition of Urdu, Nastaleeq Font*", Multi Topic Conference, 2002. INMIC 2002. International, 2002.
- [23]. Syed S. Hyder and Ali Khoujah, "*Character Recognition Of cursive scripts*", Proceedings of the 1st international conference on Industrial and engineering applications of artificial intelligence and expert systems - Vol 2, June 1998.
- [24]. J. Hébert, M. Parizeau, N. Ghazzali, "*Learning to segment cursive words using isolated characters*", Proceeding of the Vision Interface Conference, pg. 33-40.(1999).
- [25]. M. Fahmy and S. Al Ali, "*Automatic recognition of handwritten Arabic characters using their geometrical features*", Journal of Studies in Informatics and Control with Emphasis on Useful Applications of Advanced Technology, 2001.
- [26]. Syed Afaq Husain and Syed Hassan Amin, "*A Multi-tier Holistic Approach for Urdu Nastaliq Recognition*", IEEE INMIC Dec. 2002, Karachi.
- [27]. S. Snoussi Maddouri, H. Amiri, A. Belaïd and Ch. Choisy, "*Combination of Local and Global Vision Modeling for Arabic Handwritten Words Recognition*", in Eighth IWHFR, pg. 128-132. August 2002.
- [28]. Alex Cherkasov, "*Creating Optical Character Recognition (OCR) application using Neural Network*", IT toolbox Emerging Technologies Industry Articles, 27 July, 2005.
- [29]. Myriam Côté, Eric Lecolinet, Mohamed Cheriet, Ching Y. Suen, "*Building a Perception Based Model for Reading Cursive Script*", ICDAR 1995: pg. 898-901, 1995.
- [30]. Ali Ahmadi, Yoshinori Shirakawa, Md. Anwarul Abedin, Kazuhiro Takemura, Kazuhiro Kamimur, Hans Jurgen Mattausch and Tetsushi Koide, "*Real-time Character Recognition System Using Associative Memory Based Hardware*", Circuits and Systems, 2005. 48th Midwest Symposium on, August 2005.

- [31]. Roberto J.Rodrigues, Antonio Carlos Gay Thome, “*Cursive character recognition – a character segmentation method using projection profile-based technique*”, Proceedings of SCI 2000/ISAS 2000 VOLUME V, 2000.
- [32]. Ing Ren Tsang, “*Pattern recognition and complex systems*”, PhD thesis defended at University of Antwerpen on September 22, 2000.
- [33]. U. Pal and Anirban Sarkar, “*Recognition of Printed Urdu Script*”, ICDAR 2003: 1183-1187, 2003.
- [34]. A. Elgammal and M.A. Ismail, “*A graph-based segmentation and feature extraction framework for Arabic text recognition*”, in ICDAR'01, 2001,pg. 145-155, 2001.
- [35]. *Optical Character Recognition*, Available at: http://en.wikipedia.org/wiki/Optical_character_recognition, [Accessed at 17th December 2006].
- [36]. *Urdu Computing Information (Penn State)*, Available at: <http://tlt.its.psu.edu/suggestions/international/bylanguage/urdu.html#script>, [Accessed in July 2006].
- [37]. Wali, A. et el, “**Contextual Shape Analysis of Nastaliq, CRULP Annual Student Report (2001-2002)**”, p 288-302, 2002.
- [38]. *Urdu: Language of Pakistan*. Available at http://www.ethnologue.com/14/show_language.asp?code=URD [Accessed in July 2006].
- [39]. *Noori Nastaliq*, Available at <http://www.elite.com.pk/noori.html>, [Accessed at 7th January 2007].
- [40]. *The Urdu Alphabet*, Available at <http://www.unics.uni-hannover.de/nhtcapri/urdu-alphabet.html>, [Accessed at 14th January 2007].
- [41]. *Urdu*, Available at <http://www.nvtc.gov/lotw/months/february/urdu.html>, [Accessed in July 2006].
- [42]. Henry Rogers, *Writing Systems*, Blackwell publishing, 2005.

- [43]. Aamir Wali, Atif Gulzar, Ayesha Zia, Muhammad Ahmad Ghazali, Muhammad Irfan Rafiq, Muhammad Saqib Niaz, Sara Hussain, and Sheraz Bashir, “*Features of Noori Nastalique*”, Center for Research in Urdu Language Processing National University of Computer and Emerging Sciences.
- [44]. Dr. Sarmad hussain, Shafiq-ur-Rehman, Atif Gulzar, Amir wali and Jamil, “*Orthographic Analysis of Nasta’leeq Writing Style for Urdu*”, Center for Research in Urdu Language Processing National University of Computer and Emerging Sciences, 2002.
- [45]. Sarmad Hussain, www.LICT4D.asian/fonts/Urdu_nasta'le, Center for Research in Urdu Language Processing National University of Computer and Emerging Sciences.
- [46]. *Skeletonization/Medial Axis Transform*, Available at <http://homepages.inf.ed.ac.uk/rbf/HIPR2/skeleton.htm>, [Accessed in 1st December 2006].
- [47]. Zahra Shah and Farah Saleem, “*Final thesis Report*”, 2002
- [48]. Lawrence Rabiner and Biing- Hwang Juang, “*Theory and Implementation of Hidden Markov Models*” in the book, “*Fundamental of Speech Recognition*”, chapter 6, published in 1993.
- [49]. Steve Young, Gunnar Evermann, Thomas Hain, Dan Kershaw, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev and Phil Woodland, “*The HTK Book*”, December 1995.
- [50]. *A Statistical Learning/Pattern Recognition Glossary*, Available from <http://www.cs.wisc.edu/~hzhang/glossary.html>. [Accessed 17th December 2006].
- [51]. Sobia Tariq Javed, Ameera Maqbool, Sehrish Jameel and Samia Asloob Qureshi, “*Urdu Nastaleeq OCR*”, BS final year report, 2005

Appendix

Appendix A

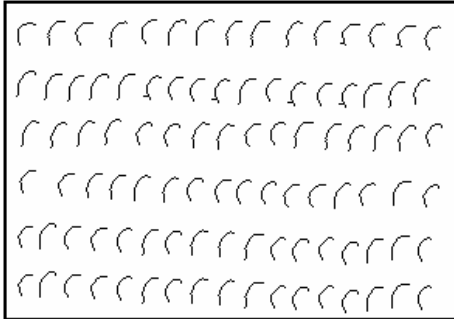


Figure 101 : Hmm00

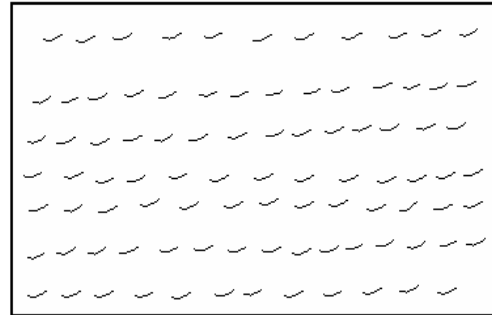


Figure 102 : Hmm01

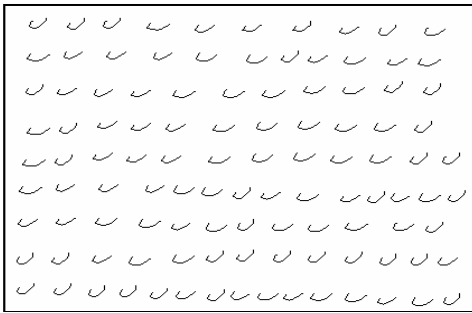


Figure 103 : Hmm02

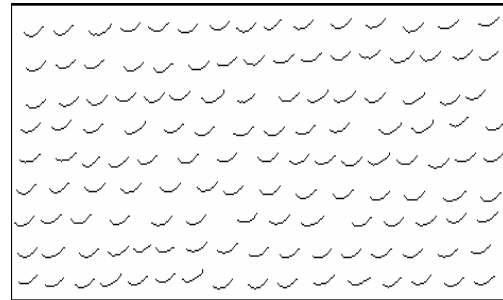


Figure 104 : Hmm03

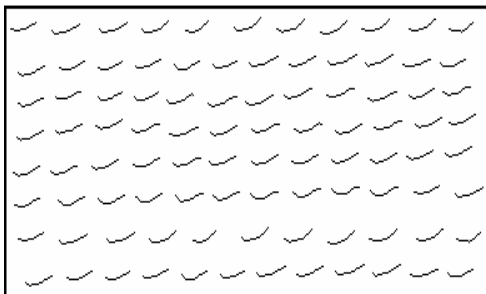


Figure 105 : Hmm04

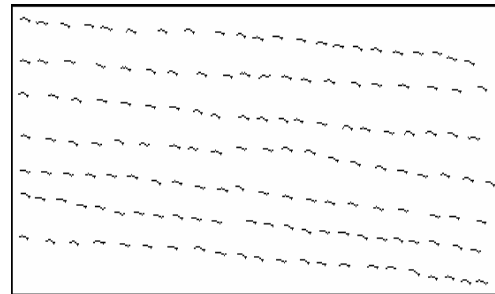


Figure 106 : Hmm05

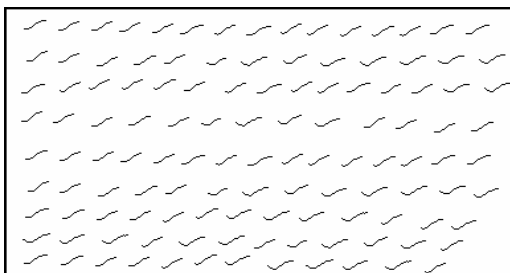


Figure 107 : Hmm06

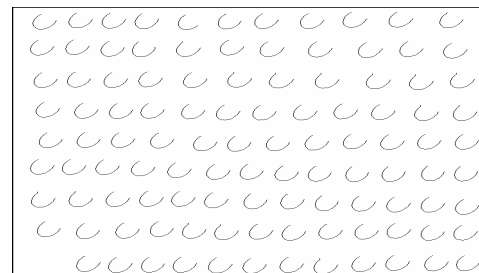


Figure 108 : Hmm07

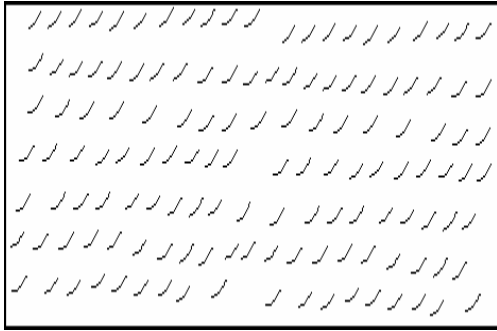


Figure 109 : Hmm08

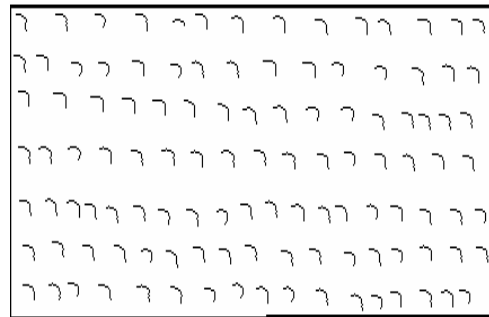


Figure 110 : Hmm09

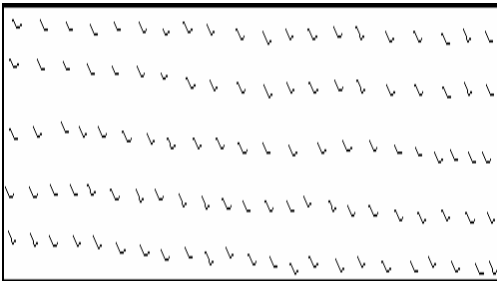


Figure 111 : Hmm10

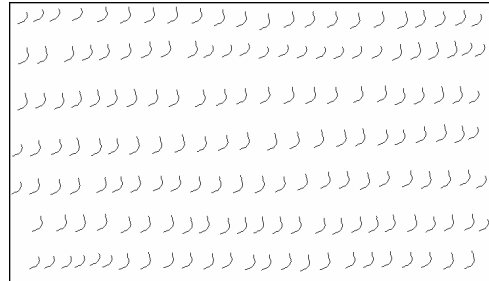


Figure 112 : Hmm12

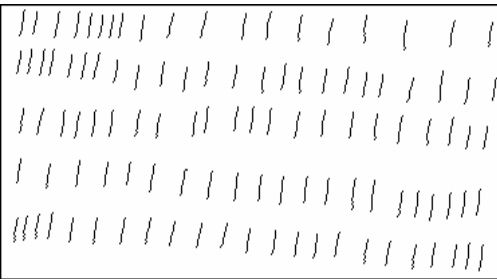


Figure 113 : Hmm013

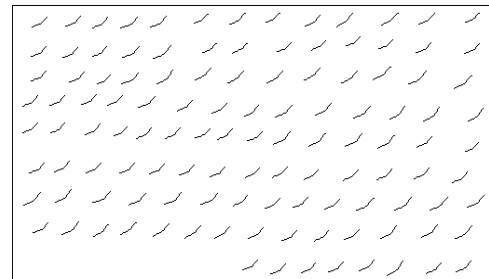


Figure 114 : Hmm014

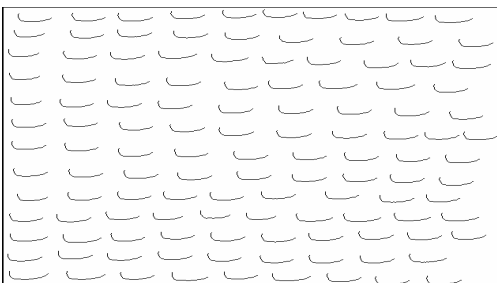


Figure 115 : Hmm015

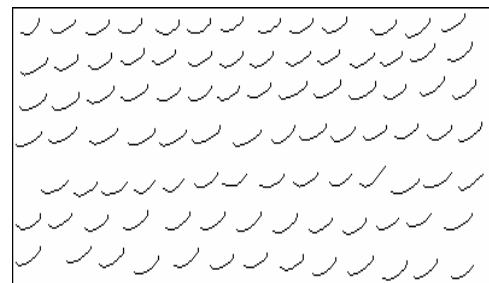


Figure 116 : Hmm016

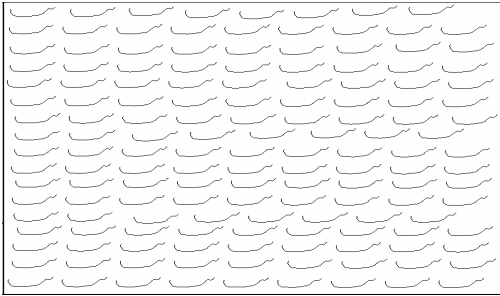


Figure 117 : Hmm017

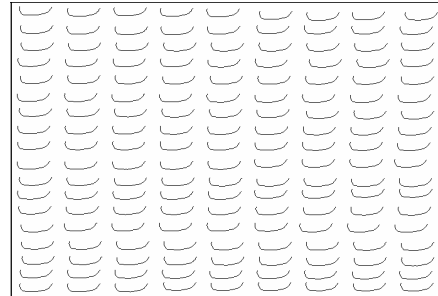


Figure 118 : Hmm018

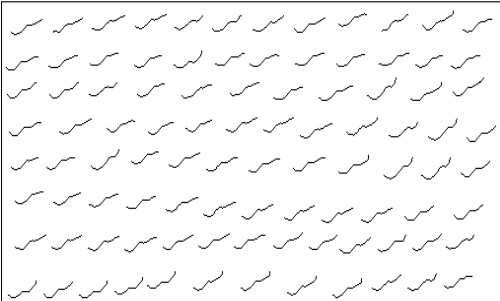


Figure 119 : Hmm019

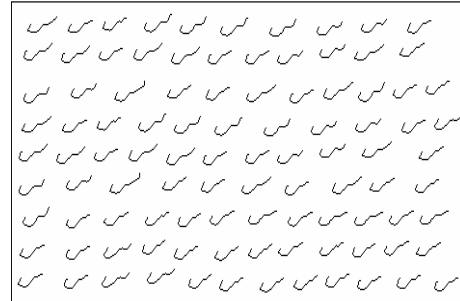


Figure 120 : Hmm020



Figure 121 : Hmm021

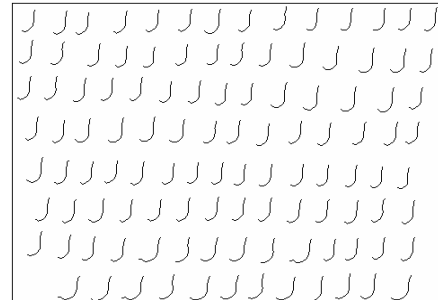


Figure 122 : Hmm022

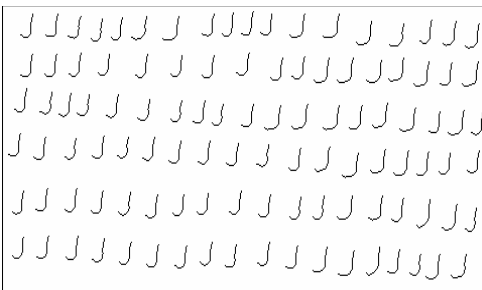


Figure 123 : Hmm023

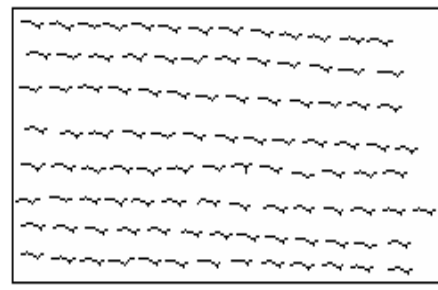


Figure 124 : Hmm024

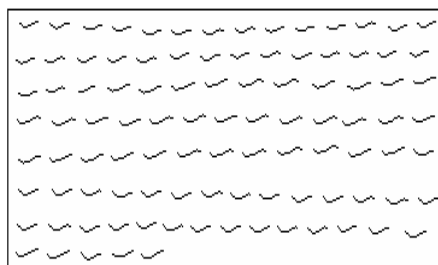
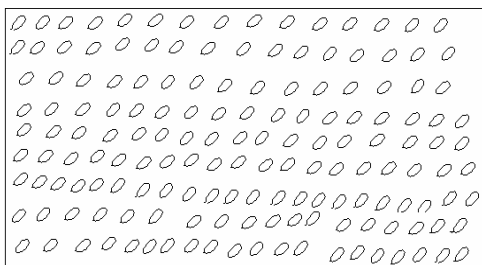


Figure 125 : Hmm025

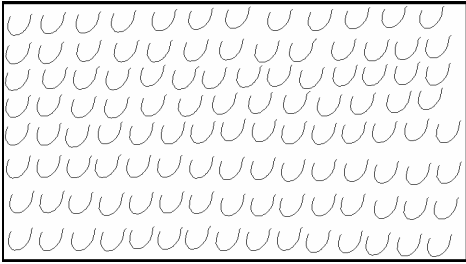


Figure 127 : Hmm027

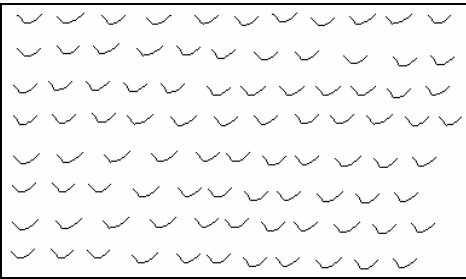


Figure 129 : Hmm029

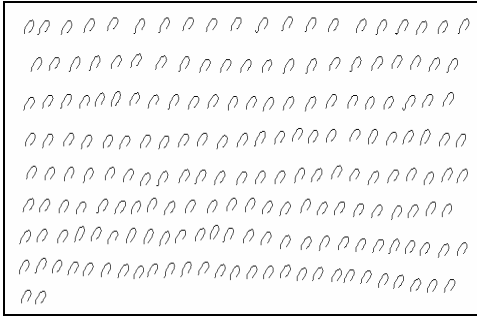


Figure 131 : Hmm031

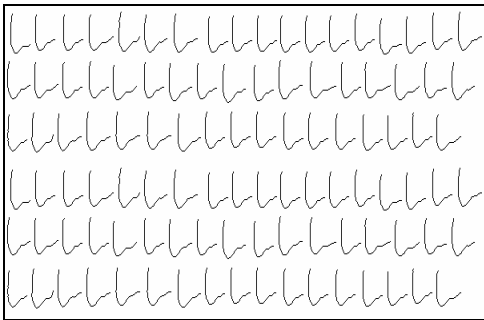


Figure 133 : Hmm033

Figure 126 : Hmm026

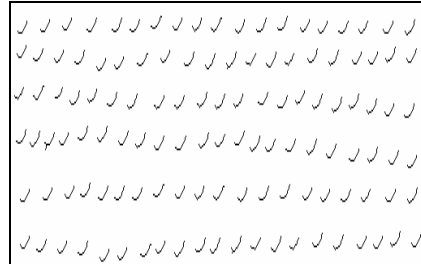


Figure 128 : Hmm028

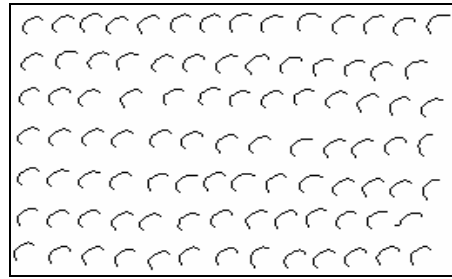


Figure 130 : Hmm030

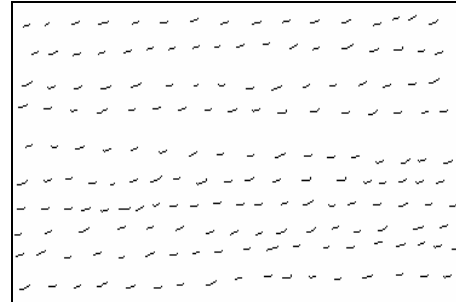


Figure 132 : Hmm032

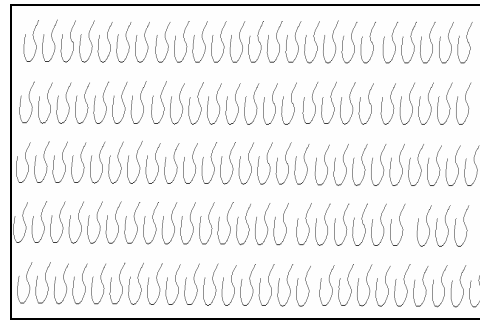


Figure 134 : Hmm034

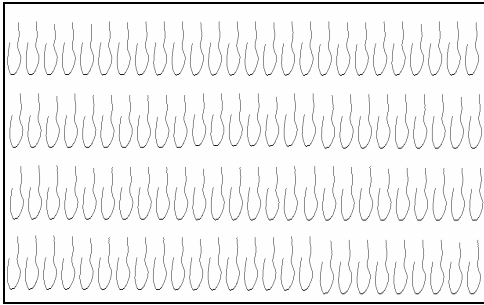


Figure 135 : Hmm035

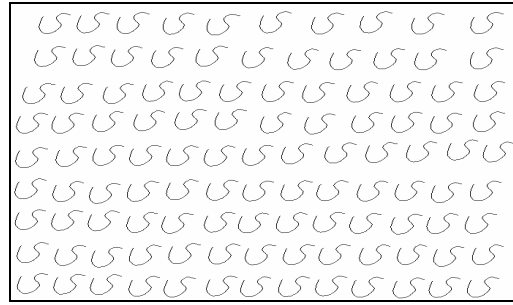


Figure 136 : Hmm036

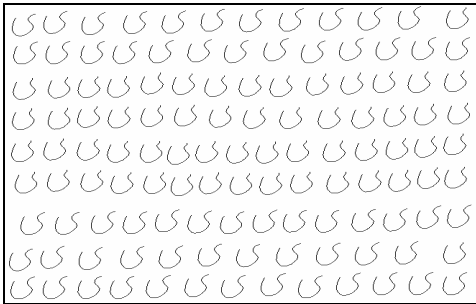


Figure 137 : Hmm037

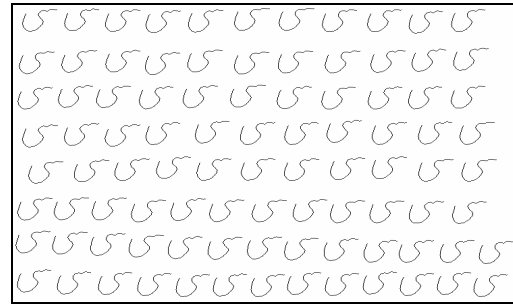


Figure 138 : Hmm038

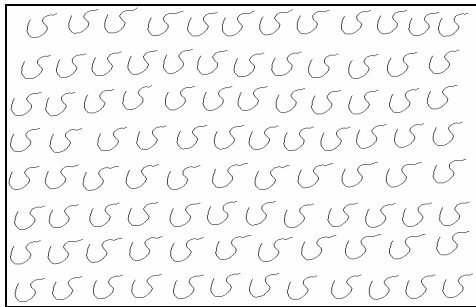


Figure 139 : Hmm039

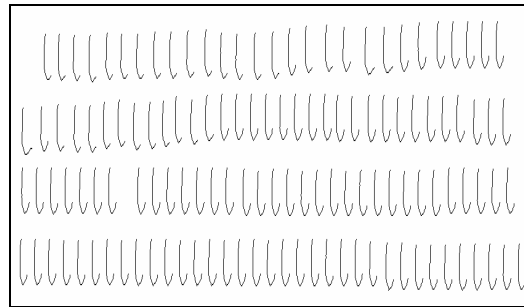


Figure 140 : Hmm040

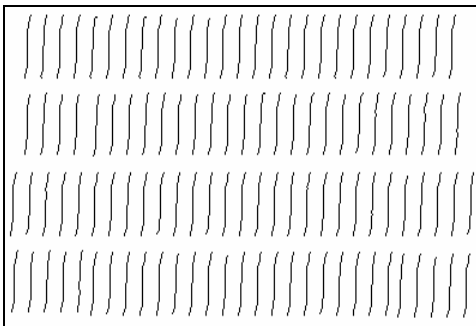


Figure 141 : Hmm041

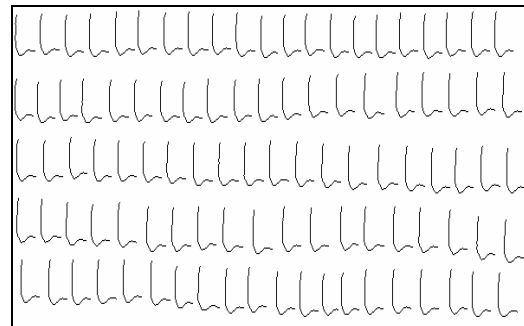


Figure 142 : Hmm042

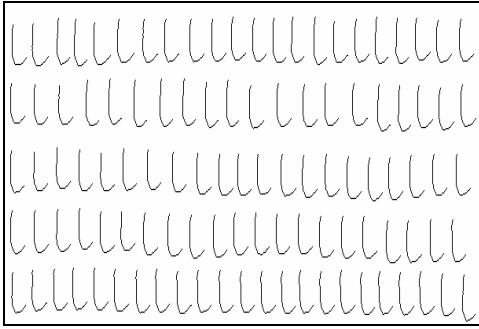


Figure 143 : Hmm043

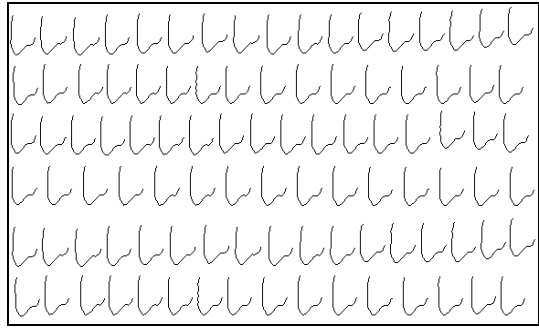


Figure 144 : Hmm044

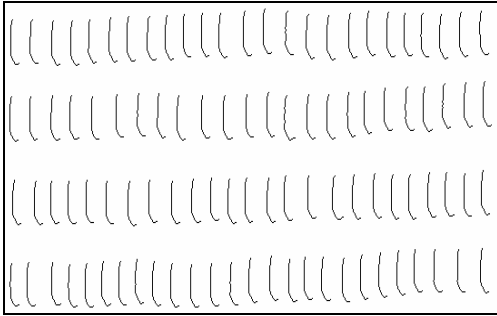


Figure 145 : Hmm045

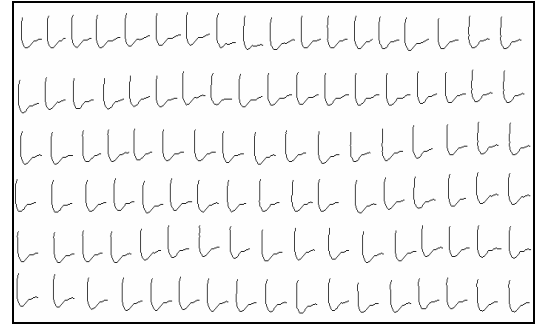


Figure 146 : Hmm046

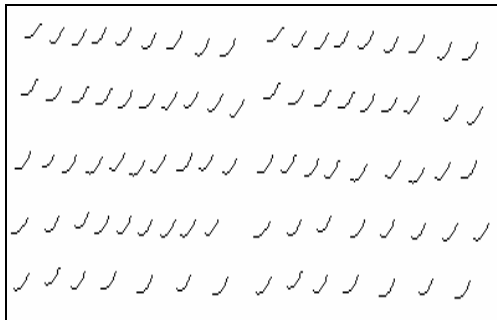


Figure 147 : Hmm047

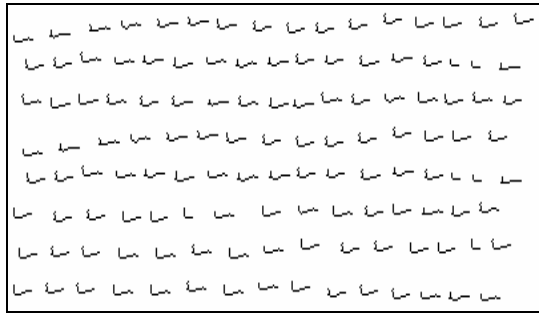


Figure 148 : Hmm048

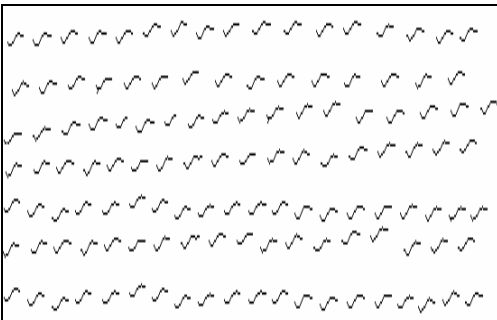


Figure 149 : Hmm049

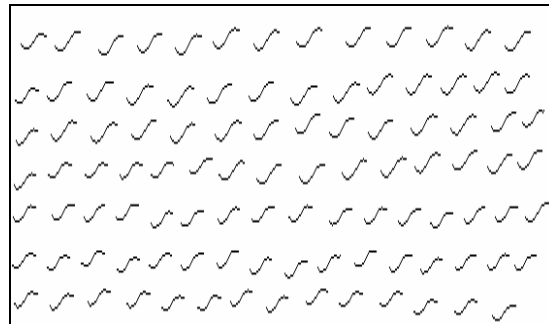


Figure 150 : Hmm050

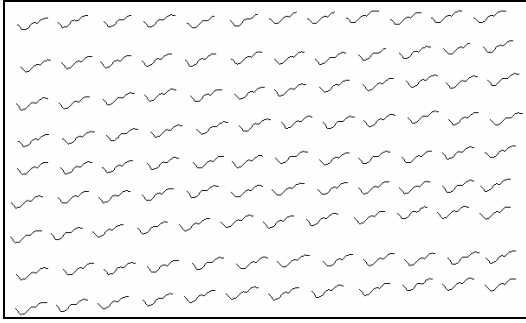


Figure 151 : Hmm051

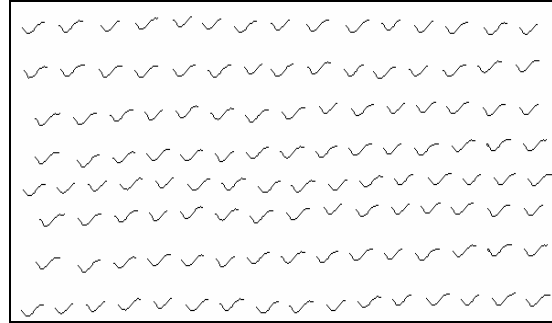


Figure 152 : Hmm052

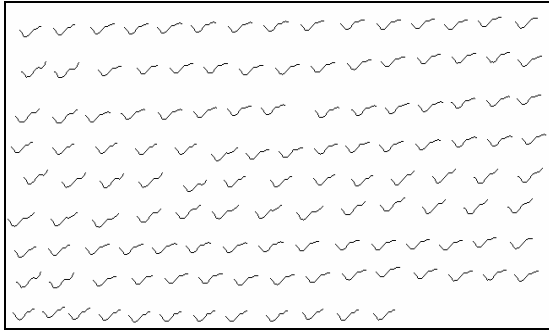


Figure 153 : Hmm053

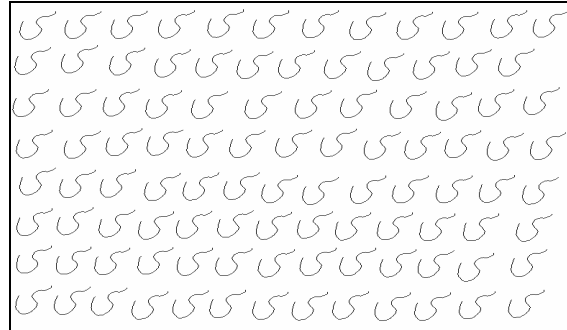


Figure 154 : Hmm054

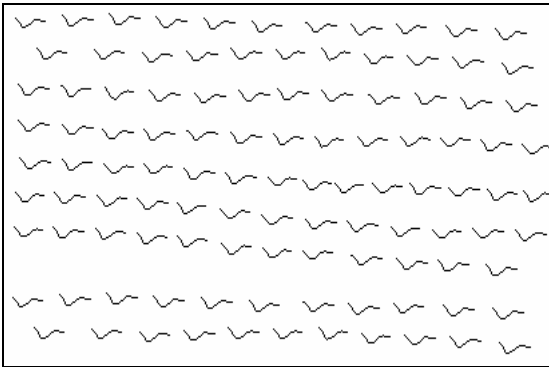


Figure 155 : Hmm055

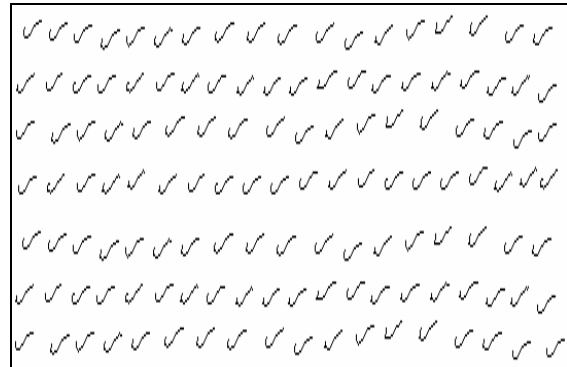


Figure 156 : Hmm056

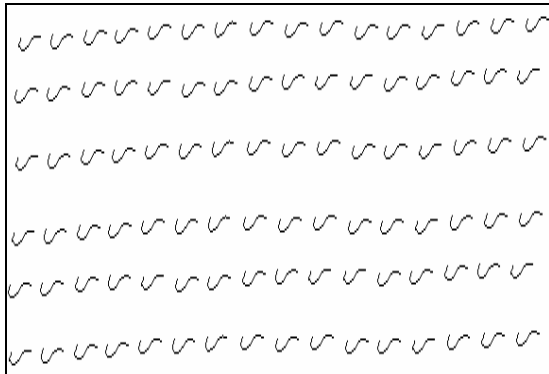


Figure 157 : Hmm057

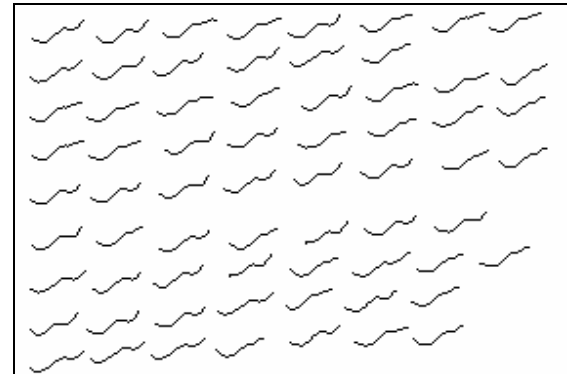


Figure 158 : Hmm058



Figure 159 : Hmm059

Appendix B

Table 20: HMM State Analysis

| Sr. | HMM Name | No. of Frames | No. of states | No. of samples |
|-----|----------|---------------|---------------|----------------|
| 1 | h00 | 3 | 5 | 100 |
| 2 | h01 | 2 | 4 | 100 |
| 3 | h02 | 3 | 5 | 109 |
| 4 | h03 | 3 | 5 | 136 |
| 5 | h04 | 3 | 5 | 100 |
| 6 | h05 | 3 | 4 | 110 |
| 7 | h06 | 3 | 4 | 122 |
| 8 | h07 | 10 | 11 | 108 |
| 9 | h08 | 2 | 4 | 137 |
| 10 | h09 | 2 | 4 | 114 |
| 11 | h010 | 2 | 3 | 180 |
| 12 | h011 | | | |
| 13 | h012 | 3 | 5 | 165 |
| 14 | h013 | 2 | 3 | 109 |
| 15 | h014 | 3 | 4 | 118 |
| 16 | h015 | 7 | 8 | 122 |
| 17 | h016 | 3 | 5 | 100 |
| 18 | h017 | 10 | 11 | 152 |
| 19 | h018 | 8 | 10 | 163 |
| 20 | h019 | 4 | 5 | 157 |
| 21 | h020 | 4 | 6 | 100 |
| 22 | h021 | 2 | 4 | 136 |
| 23 | h022 | 5 | 6 | 111 |
| 24 | h023 | 4 | 5 | 108 |
| 25 | h024 | 3 | 5 | 104 |
| 26 | h025 | 5 | 7 | 169 |
| 27 | h026 | 3 | 4 | 104 |
| 28 | h027 | 10 | 12 | 112 |
| 29 | h028 | 2 | 4 | 123 |
| 30 | h029 | 3 | 5 | 100 |
| 31 | h030 | 3 | 4 | 100 |
| 32 | h031 | 4 | 6 | 191 |
| 33 | h032 | | | |
| 34 | h033 | 9 | 10 | 104 |
| 35 | h034 | 13 | 14 | 122 |
| 36 | h035 | 14 | 15 | 104 |
| 37 | h036 | 14 | 15 | 110 |
| 38 | h037 | 12 | 13 | 122 |
| 39 | h038 | 14 | 16 | 100 |
| 40 | h039 | 14 | 15 | 100 |
| 41 | h040 | 7 | 8 | 129 |
| 42 | h041 | 5 | 7 | 112 |
| 43 | h042 | 8 | 9 | 100 |
| 44 | h043 | 7 | 8 | 107 |

| | | | | |
|----|------|----|----|-----|
| 45 | h044 | 9 | 10 | 100 |
| 46 | h045 | 6 | 8 | 100 |
| 47 | h046 | 8 | 10 | 104 |
| 48 | h047 | 2 | 4 | 130 |
| 49 | h048 | 3 | 4 | 127 |
| 50 | h049 | 3 | 4 | 126 |
| 51 | h050 | 4 | 5 | 105 |
| 52 | h051 | 5 | 6 | 108 |
| 53 | h052 | 3 | 5 | 124 |
| 54 | h053 | 4 | 5 | 116 |
| 55 | h054 | 15 | 16 | 101 |
| 56 | h055 | 4 | 5 | 108 |
| 57 | h056 | 3 | 5 | 132 |
| 58 | h057 | 4 | 6 | 100 |
| 59 | h058 | 4 | 6 | 100 |
| 60 | h059 | 2 | 4 | 126 |