Hybrid of HMM and Fuzzy Logic for Isolated Handwritten Character Recognition

Azizah Suliman
College of Information Technology
Universiti Tenaga Nasional
Malaysia

1. Introduction

Handwriting is a skill that is personal to individuals. The term "handwriting" is defined to mean as a surface consisting of artificial graphic marks conveying some message through the mark's conventional relation to language (Plamondon & Srihari, 2000). Handwriting recognition is the task of producing the symbolic form, from the stored information of the handwriting data. Handwriting data is captured and stored in its digital format either by scanning the writing on paper or by writing with a special pen on an electronic surface such as a digitizer combined with a liquid crystal display. The two approaches are respectively distinguished as off-line and on-line handwriting. On-line systems for handwriting recognition are available in hand-held computers such as PDAs with acceptable performance (Plamondon & Srihari, 2000). Off-line systems are less accurate than on-line systems due to their less informative data capturing device, which is usually the scanner. However, they are now good enough that they have a significant economic impact in specialized domains such as interpreting handwritten postal addresses on envelopes and reading courtesy amounts on bank cheques (Plamondon & Srihari, 2000).

Comparing the achievements of various researches in this field is quite difficult as the databases and general approaches might differ. Testing done with different databases would give differing results as variations and complexity of the data in the databases are not the same. Similar issues are also with approaches. Approaches would differ in recognition of characters, digits, words, cursive, non-cursive, with or without post-processing. Even though research in the area are extensive many more can be done at not necessarily in improving the percentage of accuracy but also at attempting to reduce complexity of its pre-processing techniques, its classifier, its post-processing and also the need for huge databases for trainings.

In this paper a hybrid approach of recognition is investigated with the fusion of Hidden Markov Model and Fuzzy Logic. The motivation behind this is to incorporate the syntactical nature of a fuzzy logic with the statistical approach of an HMM.

2. Handwritten Character Research

According to (Arica & Yarman-Vural, 2001) in their article that reviews the research of character recognition (CR), the CR systems have evolved in three ages. The early ages are in the period of 1900-1980. The beginning of Optical Character Recognition (OCR) was said to have started with the objective of developing reading machines for the blind. In these early systems of automatic recognition of characters, area of concentrations are either in machine-printed text or upon small set of well-distinguished handwritten text or symbols. Machine printed character recognition at that time used template matching in which an image was compared to a library of images. Statistical classifiers were mainly used for handwritten text, whereby feature vectors which were extracted using low-level image processing techniques on the binary image were fed to it.

In the second period of development in the era of 1980s -1990s, the explosion of information technology has helped a rapid growth in the area of OCR. Structural approaches were introduced in many systems in addition to the statistical methods (Belaid & Haton, 1984; Shridhar & Badreldin, 1985). The CR research was focused basically on the shape recognition techniques without using any semantic information. Although an upper limit in the recognition rate was achieved, it was not sufficient in many practical applications. Reviews of character recognition research and development during this period for off-line can be found in (Mori et.al., 1992) and in (Suen et.al., 1990) for on-line cases.

The 1990s period and onwards are referred as the advancements era, where the real progress in OCR systems is achieved. With the continuous growth in information technologies, new development tools and methodologies are utilized. In the beginning of this period, image processing and pattern recognition techniques were efficiently combined with artificial intelligence (AI) methodologies. Complex algorithms for character recognition systems were developed. With powerful computers and more accurate electronic equipments (e.g. scanners, cameras, and electronic tablets), efficient and modern use of methodologies such as neural networks (NNs), hidden Markov models (HMM), fuzzy set reasoning and natural language processing are possible. In recent systems for machine-printed off-line (Avi-Itzhak et.al., 1995; Bazzi et.al. 1999) and limited vocabulary, user-dependent on-line handwritten characters (Hu et.al., 2000; Meyer, 1995; Plamondon & Srihari, 2000) recognition rate are quite satisfactory for restricted applications. There is, however, still a long way to go in order to reach the ultimate goal of machine simulation of fluent human reading, especially for unconstrained on-line and off-line handwriting (Arica and Yarman-Vural, 2001).

2.1 Methodologies of OCR Systems

The methodologies that will be the topic of focus here are the methodologies of the off-line handwriting recognition. The sequence of approach for most of OCR systems would be to start the process from the pixel level and ending up with a meaningful text. This approach varies a great deal, depending upon the type of CR system and the methodology used. The literature review in the field of OCR indicates that these hierarchical tasks are grouped in the stages of preprocessing, segmentation, representation, training and recognition and postprocessing. In some methods, some of the stages are merged or omitted; in others a feedback mechanism is used to update the output of each stage. The three common alternative structures of word recognition systems are presented in Fig.1.

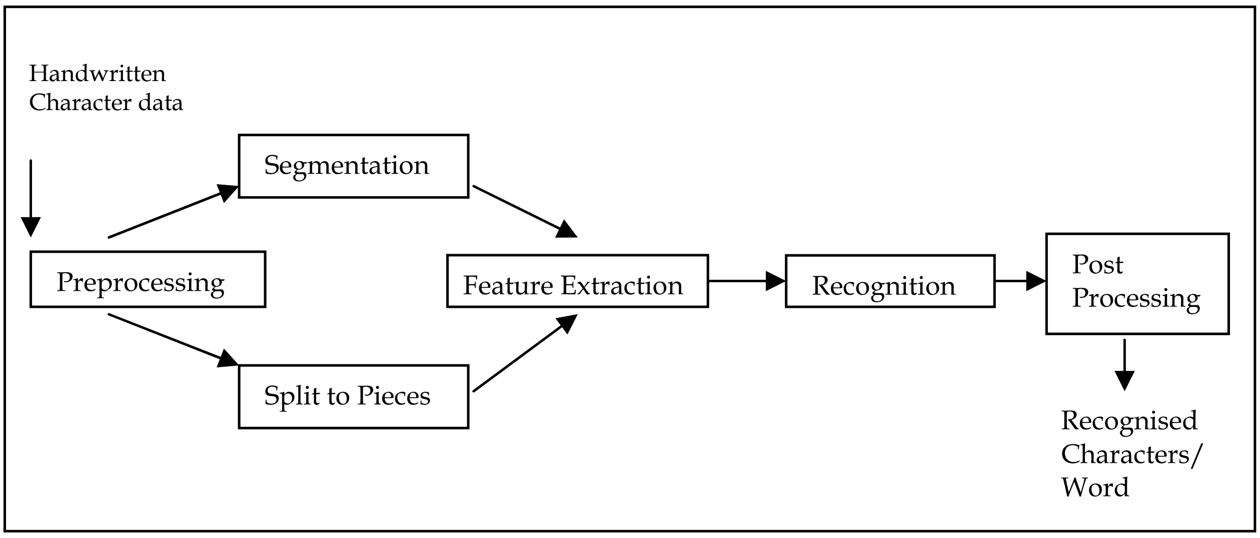


Fig. 1. Common Alternative Structures of Handwritten Characters/Words Recognition Systems

3. The Problem

In constructing systems such as a classifier, in general there are two kinds of information available: numerical information from a measuring instrument and the linguistic information from a human expert. As the problem of handwritten character recognition deals with lots of variations and complexity of data, most of the time to use a purely statistical method would be too risky. In 1965, Zadeh introduced a modified set theory namely known as fuzzy sets (Zadeh, 1965). Fuzzy logic deals with fuzzy sets that classify using unsharp boundaries. Since the data of some of the handwritten characters are sometimes vaguely distinguishable, a fuzzy inference seems to be a very logical way to deal with the recognition. Fuzzy rule-based systems utilize linguistic variables and changing numerical data of an image into its linguistic form can be very challenging. Furthermore one of the major drawbacks of fuzzy logic is the lack of learning capabilities, unlike in neural networks and HMM, where its parameters can be trained. There have been efforts in this area where neuro-fuzzy systems are introduced. HMM has been used in a lot of the handwritten character/word recognition as a classifier of characters/words and as a hybrid approach with other methods (Wierer and Boston, 2007; Gilloux et. al, 1993). In this research project, HMM will instead be use in the preparation of linguistic variables of a fuzzy rule based recogniser.

3.1 The Problem Solution

A HMM model is a very useful tool to be incorporated into a fuzzy logic rule based system. It provides an approach that is compatible to the needs of a fuzzy system. The calculation of probabilities by a statistical model such as HMM provides a solid base for the more syntactical approach of a fuzzy system. HMM yields a more accurate assessment of probabilities for the linguistic variables of a fuzzy system. However the nature of fuzziness in the data captured for the offline handwritten characters recognition research makes a pure statistical approach a little inappropriate. Fuzzy logic has been used in many of the offline researches, giving an impressive result (Bouslama, 1997; Hanmandlu, 2003). There are many ways of using fuzzy classifier into the problem of handwritten character recognition and this paper proposes a method that does not need huge training sets and that is computationally simpler.

Linguistic variables which are considered an important descriptive element in a fuzzy rule based system are prepared and trained by using HMM (Suliman et. al, 2007). Since the linguistic variables are just used at identifying strokes and curves from the input image not much training data will be needed as it would to train the HMM in identifying the strokes that made up the characters. This is in line with the motivation of this research as one of it is to minimize training data. Hence by incorporating HMM and fuzzy logic, it seems to be an idea worth investigating.

In making the research more manageable, the area of concentration is scoped down to the recognition of isolated handwritten lowercase characters. The database used is The IRESTE On/Off (IRONOFF) Dual Handwriting Database, developed by researchers from University of Nantes, France. The IRONOFF database can be obtained by contacting: Christian VIARD-GAUDIN: cviard@ireste.fr).

3.2 The Proposed System Structure

The structure of the whole system is illustrated in Fig. 2. The task of recognizing and classifying the characters from an image file, goes through a few processes as illustrated by the figure. The input image file is preprocessed using minimum preprocessing functions like binarization and thinning. The thinned image will then undergo a feature extraction process of chain-coding.

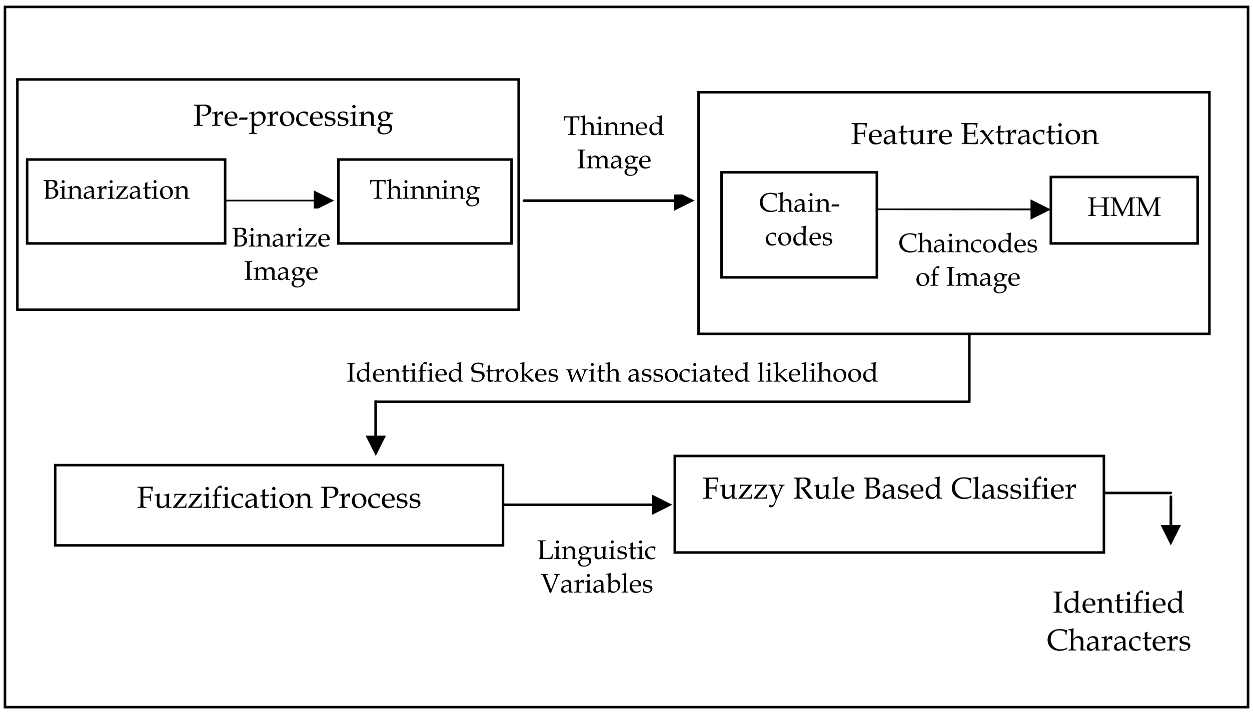


Fig. 2. The System Structure

The chain-coded image kept in a file, is then passed through a Hidden Markov Chain. The HMM will be processing the chain-codes and the output produced would be the identified strokes and its associated log-likelihood. These log-likelihood values are then converted to probabilities and pass through fuzzification process producing meaningful linguistic terms for the variables. The linguistic variables will then be used by a set of fuzzy rules to classify the character accordingly.

4. Hidden Markov Model

HMM is a doubly statistical process with an underlying Markov process that is not directly observable (hidden), but can only be observed through another set of statistical processes that produce the sequence of observed symbols (Rabiner, 1989; Kundu & He, 1991). The HMM is characterized by a finite-state Markov chain and a set of output distributions.

Following are the notations introduced by (Rabiner, 1989). The elements of the first-order HMM for character recognition are formally defined as follows.

i) *N*, the number of states in the model. Even though the states are usually hidden often there are some physical significance attached to the states or to set of states of the models.

ii) *M*, the number of distinct observation symbols per state. The observation symbols correspond to the physical output of the system being modeled. The individual symbols are denoted as $V = \{v_1, v_2, \ldots.. v_m\}$.

iii) The state transition probability distribution $A = \{a_{ij}\}$ where ,

$$a_{ij} = P[q_{i+1} = S_i \mid q_t = S_i], \quad 1 <= i,j <= N$$

iv) The observation symbol probability distribution in state j, $B = \{b_j(k)\}$, where

$$b_j(k) = P\left[v_k \ at \ t \,\middle|\, q_t = S_i\right] \qquad \begin{array}{l} 1 \le j \le N \\ 1 \le k \le M \end{array}$$

v) The intial state distribution $\pi = \{ \pi_i \}$, where

$$\pi_i = P[q_1 = S_i], \ \ 1 \le \ i \le N$$

Given appropriate values of N, M, A, B and π, the HMM can be used as a generator to give an observation sequence,

Hence, a model can be denoted by a parameter set λ = (A, B, π).
As according to (Rabiner, 1989), there are three basic problems of interest in an HMM. Problem 1 is the evaluation problem, namely given a model and a sequence of observations, how do we compute the probability that the observed sequence was produced by the model. This problem allows the model that best matches the observation be chosen. Problem 2 is where we attempt to uncover the hidden part of the model, i.e. to find the 'correct' state

sequence. Problem 3 is the one in which we adjust the model parameters so as to best describe how a given observation sequence comes about. The observation sequence used to adjust the model parameters is called a training sequence since it is used to "train" the HMM. The training problem is a crucial one for most applications of HMMs, since it allows us to optimally adapt model parameters to observed training data – i.e. to create best models for real phenomena. This is part of machine learning.

Since our problem would be an evaluation problem whereby, given a model and a sequence of observations, how do we compute the probability that the observed sequence was produced by the model or also viewed as one of scoring how well a given model matches a given observation sequence. In our view point we are considering a case in which we are trying to choose among several competing models, the model which best matches the observations. As such Problem 1 will be of our concern. Problem 3 will also be needed as we need to first train the model parameters to the observed training data. With this we would be incorporating the concept of machine learning into the system. Since the two problems will be used in this research work, both of the solutions to the problem are elaborated below.

Solution to Problem 1

As we wish to calculate the probability of the observation sequence O = O_1, O_2 ,..., O_T given the model λ, i.e. P(O | λ), a more efficient procedure to use would be the forward procedure (Kundu and He, 1991). Consider the following forward variable $\alpha_t(i)$ defined as

$$\alpha_t(i) = P(O_1 O_2 \ldots O_t, q_t = S_t \mid \lambda),$$

$\alpha_t(i)$ may then be solved inductively as follows ,

1. Initialization,

$$\alpha_1(i) = \pi_i b_i(O_1), \qquad 1 \le i \le N$$

2. Induction :

$$\alpha_{i+1}(j) = \left[\sum_{i=t}^{N} \alpha_t(i) a_{ij} \right] b_i(O_{t+1}), \qquad \begin{array}{l} 1 \le t \le T-1 \\ 1 \le j \le N \end{array}$$

3. Termination :

$$P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i),$$

In modelling the problem to an HMM, the observation sequence would be the extracted chain-codes of the hand-written characters images. The chain codes will then be fed to the appropriate HMM of strokes to be identified. The codes will be 'churned' by the HMM and as a result strokes will be identified together with its associated probabilities. The strokes as identified will go through a fuzzification algorithm that will change it to linguistic variables which will then be utilized by a fuzzy classifier.

Solution to Problem 3

The third problem is the most difficult as it is to determine a method to adjust the model parameters (A, B, π) to maximize the probability of the observation sequence given the model. There is no known way to analytically solve for the model which maximizes the probability of the observation sequence. However we can choose λ = (A, B, π) such that P(O | λ) is locally maximized using an iterative procedure such as Baum-Welch method or equivalent. Here the iterative procedure of Baum-Welch will be discussed as a solution for problem 3.

To implement the solution we will first need to define the variable $\gamma_t(i)$ the probability of being in state S_i at time t and then define $\xi_t(i,j)$ the probability of being in state S_i at time t and state S_j at time *t+1*, given the model and the observation sequence, i.e.:

$$\gamma_t(i) = P(q_t = S_i \mid O, \lambda)$$
$$= \frac{\alpha_t(i)\beta_t(i)}{P(O \mid \lambda)}$$

$$\xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_i \mid O, \lambda)$$
$$= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O \mid \lambda)}$$

Now we have,

$\sum_{t=1}^{T-1} \xi_t(i,j) =$ expected number of transition made (from $S_i \; to \; S_j$)

$\sum_{t=1}^{T-1} \gamma_t(i) =$ expected number of transition from S_i.

The Baum-Welch re-estimation formulas for *A,B* and *π* are

$$\bar{\pi}_i = \gamma_1(i)$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$\bar{b}_j(k) = \frac{\sum_{\substack{t=1 \\ s.t. O_t = v_k}}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

Iterative application of these formulas will converge to a local maxima of $P(O \mid \lambda)$.

5. Fuzzy Logic

Fuzzy logic refers to all of the theories and technologies that employ fuzzy sets, which are classes with unsharp boundaries. Not as in a classical set theory, the concept in fuzzy sets does not have a well defined natural boundary. A representation of the concept closer to human interpretation is to allow a gradual transition. In order to achieve this, the notion of membership in a set needs to become a matter of degree. This is the essence of fuzzy sets.

A fuzzy logic system would usually consist of the following:

a) A fuzzification unit which maps the measured inputs, which might be in the form of crisp values, into the fuzzy linguistic values used by the fuzzy reasoning mechanism.

b) A knowledge based which is the collection of the expert control rules (knowledge) needed to achieve the control goal.

c) A fuzzy reasoning mechanism which performs various fuzzy logic operations to infer the control action for the given fuzzy inputs.

d) A defuzzification unit which converts the inferred fuzzy control action into the required crisp control value.

In this research it is to the first part of the fuzzy logic system that an HMM Model is introduced to. The fuzzy linguistic values are extracted and quantified by an HMM Model. As previously mentioned, the extracted chain-codes of the hand-written character images will then be fed to a few HMM models of strokes to be identified. As a result, strokes will be identified together with its associated log likelihoods which may in turn be used derive to probabilities. The strokes as identified will go through a fuzzification algorithm that will change it to linguistic variables which will then be utilized by a set of fuzzy rules to determine the class of the character.

6. The Pre-processing Phase

The raw data of handwritten characters, no matter how it is acquired, will be subjected to a number of preprocessing steps to make it useable. The preprocessing phase aims to extract the relevant textual parts and prepares them for segmentation and recognition. The main objectives of preprocessing are i) noise reduction, (ii) normalization of data and (iii) compression in the amount of information to be retained. In noise reduction alone there are hundreds of available techniques which can be categorized into three major groups of filtering, morphological operations and noise modelling (Serra, 1994; Sonka & Boyle, 1999). Filters can be designed for smoothing (Legault & Suen, 1997), sharpening (Leu, 2000), thresholding (Solihin & Leedham, 1999), removing slightly textured background (Lee & Fan, 2000) and contrast adjustment process (Polesel et. al., 1997). Various morphological operations can be designed to connect broken strokes (Atici & Yarman-Vural, 2001), decomposed the connected strokes (Chen et. al., 1994), smooth the contours, prune the wild points, thin the characters (Reinhardt & Higgins, 1996), and extract boundaries (Yang & Li, 1995).

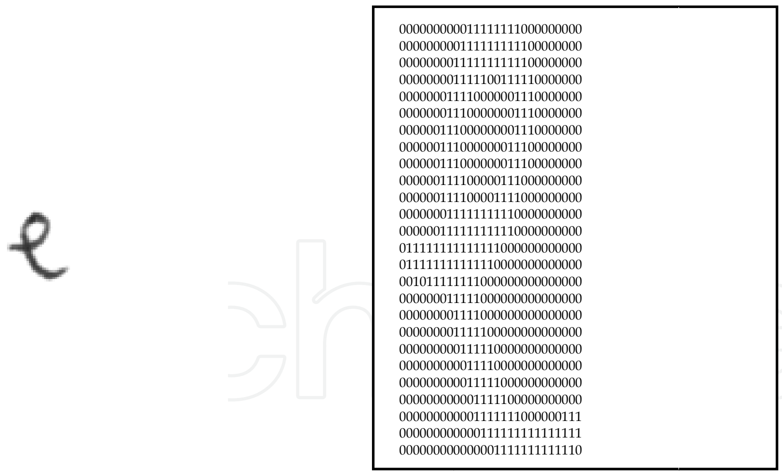
In this research work a minimal number of preprocessing processes are used. The preprocessing steps are shown in Fig. 3. An image file of handwritten character will first be read and binarized. Then reference line of upper and lower base line of the character will be estimated. The estimation of the upper and lower base line will be used to classify the characters into its three groups of either ascenders (e.g. *h, l, t, f, d, b*), descenders (e.g. *g, p, q, y*) or neither (e.g. *a, c, e, i, m, n* etc.) . The subsequent sections will explained further the pre-processing phase undertaken in this research.

Pre-processing Steps	
Step 1:	Binarize the character image. Input: An image file of the character in .bmp format Results: A matrix of the character image in 0's (for background) and 1's (for the foreground that makes up the character contour).
Step 2:	Estimate Reference Line to classify the three groups of characters (ascenders, descenders, neither). Input: The character matrix from Step 1. Results: A classification of the character into its 3 groups. To be used as one of the character features in the fuzzy inference system.
Step 3:	Thinned the image into edges of one pixel thick. Input: The character matrix from Step 1. Results: A character matrix with all (or at least most of) its edges with one pixel thickness only.

Fig. 3. Pre-processing Steps

6.1 Binarization

When an image is captured, it is stored in the form of pixel density value, which means each pixel has the value of a 0-255 (for a gray scaled image). Many researchers choose to work with a binarize image where all the grey value will be threshold and converted to either 0 for white (background) and 1 for black (foreground). This process is known as binarization.
The method used to binarize is known as thresholding. In thresholding, the gray-scale or color images are represented as binary images by picking a threshold value. The two categories of thresholding are global and local thersholding. In global thresholding, one threshold value is used for the entire document image which is often based on an estimation of the background intensity level with that of the image using an intensity histogram (Chen et. al., 1994). Local or adaptive thresholding use different values for each pixel according to the local area information (Saula & Pietikainen, 2000). Local thresholding is commonly used in works that involve images that are of varying level of intensities, such as pictures from satellites cameras or medical scanned images. For simple images like handwriting, where the characters are written on a white background, using a global threshold would suffice to distinguish the background and the foreground. Fig. 4 displays an image of the character 'e' in its image file format of .bmp and the same character after it had been binaries and save in a text (.txt) file.



```
00000000001111111000000000
00000000011111111100000000
00000000111111111100000000
00000000111110011110000000
00000001111000000111000000
00000001110000000111000000
00000011100000000111000000
00000011100000001110000000
00000011100000001110000000
00000011110000011100000000
00000011110000111100000000
00000001111111111000000000
00000011111111110000000000
01111111111111100000000000
01111111111111000000000000
00101111111110000000000000
00000001111100000000000000
00000000111100000000000000
00000000111110000000000000
00000000011111000000000000
00000000001111000000000000
00000000001111100000000000
00000000000111110000000000
00000000000111111000000111
00000000000001111111111111
00000000000000111111111110
```

Fig. 4. The character “e” in its image file format (.bmp) and in its binarize text file format

6.2 Reference Line Estimation

One preprocessing technique that has been particularly helpful in determining features in a word is reference line estimation. (Bozinovic & Srihari, 1989) referred to it as, the task of locating such lines as the lower line, lower baseline, upper baseline and the upper line. To determine these lines, an approach based on that proposed by (Bozinovic & Srihari, 1989) is considered. The approach however was for a word image and in such images the reference lines are more distinguishable as the input are larger.

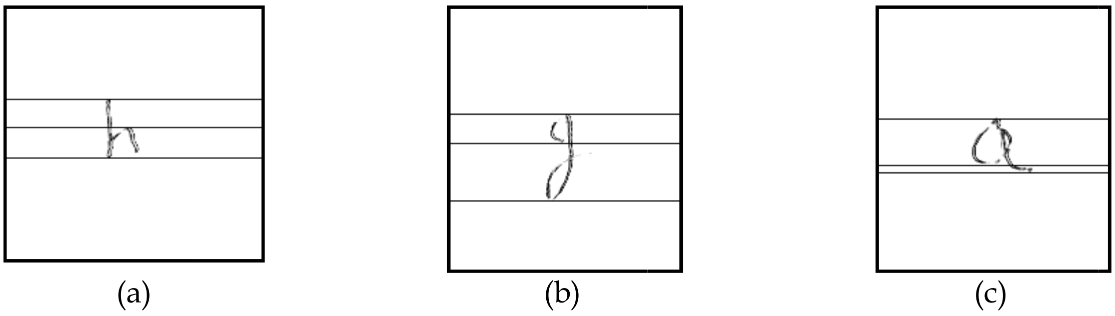


Fig. 5. Examples of characters after line referencing (a) an ascender, (b) a descender, and (c) neither an ascender nor descender.

With a little improvisation the similar approach was implemented for character image. Examples of the line referencing used in this research are shown in Fig. 5. The first step would be to generate a horizontal density histogram for the character image. This was done based on a black pixel count in each row (horizontal direction at each position on the y-axis). From the top, the first row count that have more than a minimum threshold (in this case a pixel count of 2 is used) and an upper line is found. Next, traversing from the bottom of the image or the last row in the image, the first row count that is more than the minimum threshold would indicate the lower line. Estimating the middle line is slightly more complex. First we have to access roughly if the image is denser in lower or the upper zone. This is estimated by first determining the mid line, i.e. the average of upper line and lower line. Then the pixels density of the upper and the lower zones are calculated. If the upper zone is denser then we start traversing from the mid line to the upper line, otherwise we traverse from the mid line to the lower baseline. Taking a threshold T, that is the average density of the character, we scan through density vector and the first line that have a density value that is lower than T, would be consider as the middle line. The middle line produced was a better option than the mid line gotten from the average of the lower and the upper lines. This estimate of the middle line was favourable enough for the next step.

```
Gap1 = upper line – middle line
Gap2 = middle line – lower line

If Gap1 is small or Gap2 is small
    Alphabet is Neither
else
  if density in upper zone > density in lower zone
          Alphabet is Descender
Else
          Alphabet is Ascender
```

Fig. 6. Algorithm for grouping characters

Once the upper, middle and lower lines had been placed, the character will be heuristically categorized into an ascender, a descender or neither. The algorithm for the simple heuristic technique used in the research is given Fig. 6. The grouping of the characters into the 3 groups are utilized directly by the recognition phase and if grouped correctly it would have scoped down the classification phase from the 26 possibilities (26 alphabets of a .. z) to about a third lesser of possibilities.

The heuristic technique was compared to an HMM model created to group the character using the vertical density of the image. Relying on the fact that some of the characters in a certain group would be denser in the upper zone than the lower zone and so on, the result of the grouping using the heuristic method is given in Table 1. The heuristic technique yield a correct grouping rate of 70.35% while the HMM technique yields a correct grouping rate of 64.3%.

	Classified group			
Actual Group	*Neither*	*Ascenders*	*Descenders*	*Total*
Neither	***94.39***	*3.57*	*2.04*	*100.0*
Ascender	*21.43*	***63.09***	*15.48*	*100.0*
Descenders	*27.38*	*19.05*	***53.57***	*100.0*

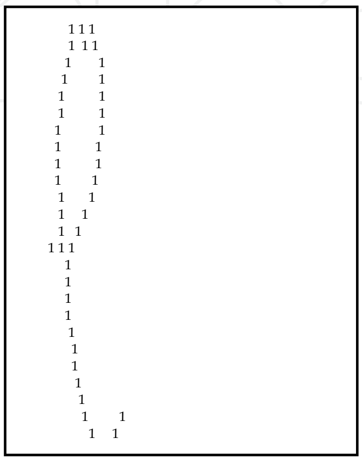
Table 1. Confusion matrix of the three groups of characters

6.3 Thinning

There are a number of algorithms available for the thinning process. Thinning algorithms has to satisfy among others the following two constraints:

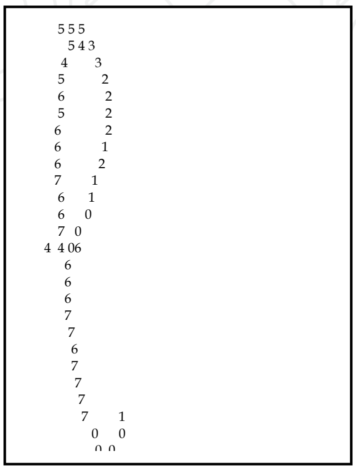
- Connectivity must be maintained at each iteration. Removal of border pixels must not cause discontinuities.
- The end of the thinned shape limbs must not be shorten.

Since the image of handwritten words are relatively small and less elaborate and less convex than an image of other objects, a slightly simpler version of a thinning algorithm would suffice. The thinning process used in this research is a one pass thinning algorithm that has been slightly modified to ensure connectivity and at the same time to repair any repairable discontinuities. An example of a binarize image of the same character 'e' as in Fig. 4, which is before thinning procedure, is shown in Fig. 6 after it had been thinned. The chain-coded version of the same 'e' is in Fig. 7 and will be explained in the next section.



```
    111
    1 11
   1    1
  1     1
 1      1
 1      1
1       1
1      1
1      1
1     1
 1   1
 1  1
 1 1
111
  1
  1
  1
  1
   1
   1
   1
    1
    1
     1    1
      1  1
```

Fig. 6. A thinned binarize image of 'e'



```
  555
    543
   4    3
  5      2
  6       2
  5       2
 6        2
 6       1
 6      2
 7     1
  6   1
  6  0
  7 0
4 406
   6
   6
   6
   7
    7
     6
     7
      7
       7
       7     1
          0   0
           0 0
```

Fig. 7. An image of chain-coded "e"

With the process of thinning where pixels are removed, sometimes the image can get broken or disjointed. To cater for such cases, images that had been thinned will be passed through a special function that would check and repair simple disjointed edge that was caused by thinning.

7. The Feature Extraction Phase

The feature extraction phase in a handwriting recognition system is agreed by many (Suen, 1986; Impedovo et. al., 1991; Arica & Yarman-Vural, 2001) to hold a very important role. Feature extraction can be defined as the process of extracting distinctive information from the matrices of digitized characters. In OCR applications it is important to extract those features that will enable the system to discriminate between all the character classes that exist. A suggested reading on the survey of feature extraction methods for handwriting recognition would be (Oh et. al. 1999).

Geometrical and topological representation is one of the many feature extraction methods used in the OCR research. This method has proven to be the most popular feature extraction method amongst researchers (Blumstein & Verma, 1999). This type of representation is able to encode some knowledge about the structure of the object or may provide some knowledge as to what sort of components make up that object. There are hundreds of topological and geometrical representations than can be grouped to a few categories (Arica & Yarman-Vural, 2001).

One popular category in geometrical and topological representation of features is by extracting and counting topological structures. Common primitive structures that are searched from a character or word image are strokes which may be as simple as lines and arcs or as complex as curves and splines. Characters and words can be successfully represented by extracting and counting many topological features such as the extreme points, maxima and minima, cusps above and below a threshold, openings, cross points (x), branch points (T), line ends, loops and many more (Madhvanath & Govindaraju 1999; Madhvanath et. al., 1999).

Coding is a category where the strokes of the character are mapped into chain-codes. One of the most popular coding schemes is Freeman's chain code (Freeman, 1974) even though there are many versions of chain coding. Fig. 9. shows a directional guide of a Freeman Code. The following section will discuss the process of chain coding as used in this research work.

Once the image has been pre-processed and chain-codes of the character to be classified had been stored, it is now ready for the feature extraction phase. The feature extraction phase is just as important as the classification phase. In this research work the feature extracted will be used solely as the input for the classification phase. This component is regarded as very important because the focal point of the research lies in the success of the features extracted. Fig. 8 gives an account of the steps in the feature extraction phase.

Feature Extraction Steps	
Step 1:	Decode the one-pixel thinned image into chain codes using the Freeman code of directions.
Step 2:	Examine the segments of chain codes that made up the character, to decide which HMM Models to call.
Step 3:	Call the appropriate HMM Models for the segments. This will produce log-likelihood values of the strokes identified. Take the possibilities of all strokes that are above its lower threshold value.
Step 4:	Change the log-likelihood values of the strokes into numbers between 0 to 1 to reflect the probability values.

Fig. 8. Feature Extraction Steps

7.1 Chain coding

The one-dimensional model of the image is obtained by tracing the contour edges of the character image and representing the path by Freeman chain codes. The objective of the edge tracing would be to get chain codes that would traverse an image of a handwritten character as naturally as it would as it was written. The challenge of the chain-coding process would lies very much on the way the image would be traverse and the starting point of the traversing method. A same image will produce a different chain-code if it starts from a different point or traverse in a different direction. Consistency is required in order to minimize variations in chain-codes of the same character.

The image is traversed using the connected component analysis algorithm. It then performs a traversal of the skeleton, segmenting it into strokes separated by points that have one or more than three neighbours (since these points are either endpoints or junctions where different strokes meet). The general steps followed in traversing the image are given in Fig. 10. The algorithm is implemented as a recursive function in C codes. It will traverse a body of connected components recursively and return to the calling function once all the pixels in the connected component have been cleared. It will be called again if there are still connected components left in the image.

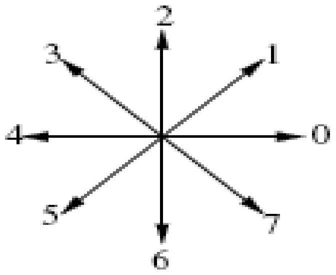


Fig. 9. Freeman's Directional Guide

```
While there are still edges to be traverse
Begin
     Find the starting point to traverse.
     From starting point till junction point or end  point
    Begin
                 Find and store the direction code of the point
                 by investigating its neighbour position.
               Once done, cleared pixel from image.
     End
End
```

Fig. 10. Algorithm for Traversing the Image

An example of an image of character "e", the same character used in Fig. 4 and Fig. 6., is shown in Fig. 7, after it has been chain coded. From the chain codes of the character, features such as the type of strokes will be extracted. The strokes will be identified with a set of probabilities. How these features are extracted by means of HMM and later used by a fuzzy inference system for classification is the other novelty of this research work.

7.2 Extracting Features from Chain-codes

After a careful study of the different types of stroke directions in a character set a few types are selected and thought to have distinguishing factors to identify a character class. Table 1 shows the types of strokes that are used to distinguish the characters. In order to simply the strokes without needing to consider the way it might be traverse, codes that visually produced the same, the directions are usually combined. For example, in identifying a vertical stroke, if the image is traverse from bottom up then the direction will be North and if it is traverse top to bottom the directions will be South. Of course the traversing method is meant to be consistent and it will always traverse the same way, but just in case, the directions are disregarded and both will be considered as a vertical stroke. Hence are true for the Horizontal stroke, the Right Slant and the Left Slant.

The other two obvious strokes are named as the C-curve and the D-curve. The last two strokes will be given the highest priorities in the investigation of strokes. In the case where the curve might be broken and not too distinguishable then the normal NSEW directions will be investigated. The combination of these smaller identifiable strokes can still make a good set of features for producing linguistic variables. The C-curve would be a prominent feature in characters like a, c, d, e etc., and the D-curve in characters like b, p, etc. Some of the characters may have the combinations of both curves and some may have a fuller curve than others. All these differences are projected in the linguistic that will describe the strokes.

The codes that form the strokes and the visual examples of some of the strokes investigated are shown in Table 2 and Fig. 11.

Types of Strokes Identified	Corresponding Direction Codes
Horizontal lines	4 or 0
Vertical lines	2 or 6
Right Slant	1 , 2, 0 or 5, 4 , 6
Left Slant	3, 2, 4 or 7, 6, 0
Loop	6, 5, 4, 7, 0, 1, 2, 3
Right Hook	6, 7, 0 1
Left Hook	6, 5, 4, 3
C – curves	7, 6 , 4, 5 or 1, 2, 4, 3
D-curves	5, 6, 0, 7 or 3, 2, 0, 1

Table 2. Among the types of strokes identified from the character image

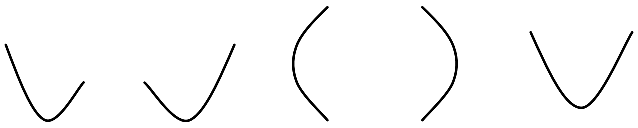


Fig. 11. Visual examples of some of the strokes identified (from left to right: Right hook, Left hook, C-curve, D-curve, U-curve).

From the chain codes of the character, features such as the type of strokes will be extracted. The strokes will be identified with a set of probabilities. HMM as mentioned before will be used for this purpose. In addition to the standard left-right model, null transitions are used for state skipping, which allows the model to tolerate omissions of some features, whereas self loops are used for modelling the repetition of features. Most of the strokes shared the same model topology. It was tested during development of the models, that increasing the number of states in the model does not improve the probabilities too much. This is probably because the strokes are all relatively very simple. However it is found that the best is to have models for as many strokes variation rather than having one and conclude its variations from the emerged log-likelihood values. For example, rather than have one model for vertical line, we tend to have a few such as for right slant, left slant, very tall vertical and small vertical lines. Since types of stroke played a distinguishing factor in the classification phase, its identification is considered very important.

The major drawback of using chain codes as the features for the HMM model is the observation symbols (which are the Freeman direction codes) are usually common and appear in more than one stroke models. One very important observation made during the development of the HMM models for the various strokes is, it is difficult to distinguish strokes that have similar observation symbols. If we were to pass the chain codes for Vertical_Lines to both the C_curve's HMM Model and the Vertical_Lines's HMM Model, it will be recognized by both and sometimes the log-likelihood values alone would not be enough to ascertain which model fits the chain codes better. In order to minimize confusion that might occur in the identification of the strokes, the of list chain codes to be processed would first be scanned through for two purposes. The first intent is to clean the list from any spurious direction codes that might appear in the strokes. This is done by removing any symbols that appear only once in a list of chain codes that is longer than 5. The second purpose is to determine which HMM model to be invoked and thus reduce the degree of confusion to ascertain the types of strokes through its log-likelihood values alone. The general algorithm for the above mentioned tasks is given in Fig. 12. The decision to determine which HMM models to be called is by examining the number of different symbols that occurred in the chain codes. For a list of chain codes that have the occurrence of 4 or more different symbols, it is assumed that the stroke are complex and as such it will fit the description of a curve or loop, and so the corresponding HMM model would be called. For a list of chain codes that is made of less than 4 symbols would be considered as simple strokes and the appropriate model will be called.

```
Procedure for Determining Which HMM Model to invoke

        Create and initialize an array to hold the count for each of the 8 possible
        observation symbols in a list of chain codes

        Count the occurrence of each symbol in the list of

        Clean the list from any spurious symbols that might appear in the chain codes,
        i.e. any symbols that only occur once in a list that is longer than 4 will be
        removed.

        Count the number of different symbols that appear in the chain codes

        If (Count is more than 3)
                Call HMM Models for curves and loop
        Else
                Call HMM Models for N,S,E,W and other simpler strokes
```

Fig. 12. Algorithm to Determine HMM Models to be invoked

7.3 Preparing Input for Linguistic Variables

A linguistic variable enables its value to be described both qualitatively by a linguistic term (i.e. a symbol serving as the name of a fuzzy set) and quantitatively by a corresponding membership function (which express the meaning of the fuzzy set). Linguistic term is used to express concepts and knowledge in human communication, whereas membership function is useful for processing numeric input data.

In this research the classification component of the system which is based on the concept of fuzzy logic is developed purely using Mathlab Toolbox of Fuzzy Logic. This component will be discussed in the next section. Since we will be utilizing Mathlab for this purpose all the fuzzification process of the linguistic variables and the inference for the classification are just a matter of invoking the functions in the toolbox. The fuzzy inference system, or FIS as it is known in Mathlab, however accepts probabilities of the input variables in order to change it to its corresponding linguistic variables. Since the HMM models of the strokes, give as an output the log-likelihood of the strokes fitting the model λ = (A, B, π), it is our task to change the log-likelihood values to a value resembling probability (value between 0 and 1). The log-likelihood values, which computation is based on a series of intensive calculations of forward and backward variables and then re-estimation in the training part, and this values will then be used in the calculation of the forward variable again in the testing part inclusive of scaling in order to avoid the values going too small and to infinity, would usually gives a value that is very unlike a probability. In order to change these values to more meaningful probability values a simple method is employed.

During testing, the most perfect and the least perfect strokes of each type will be carefully selected and tested. The term "perfect" in describing any type of strokes would be a stroke that is most prominent in visibility of the image to resemble the contour of the stroke in question. For example for a vertical line (South), that would be the tallest and the most vertical of lines, such as in the character "p" or "d" or "l" and so on. The description of "least perfect" would be strokes that have minimum features to fit the category, e.g. in this example would be a very short vertical line. The log-likelihood value of the perfect stroke, named it x, would be used as an upper threshold and the log-likelihood value of the least perfect stroke, named it y, will be used as a lower threshold, i.e. the threshold of acceptance.

Log-likelihood of the current stroke being investigated name it m,

To illustrate the idea further, assumed the example below.

Assuming the following chain codes of a south stroke (6 6 6 6 6 7 6 6 6 7 6 6 6 7 6 6 7), yields a log-likelihood of, named it m, -18.357026. Assume x = -31.981509 and y = -3.427863. So, the probability of the chain codes above fitting the South model would be,

$$\frac{m}{y} = \frac{-18.357026}{-31.981509} = 0.5739887$$

The threshold for acceptance of any strokes for this category would be,

$$\frac{y}{x} = \frac{-3.427863}{-31.981509} = 0.1071826$$

As such, anything below this threshold value would not be accepted. Of course this upper and lower threshold values differ from stroke to stroke. These probability values need only to be fed to the FIS designed for the classification task for it to be fuzzify into its corresponding linguistic variables and be used in the classification task.

8. Classification Phase

A HMM model is a very useful tool to be incorporated into a fuzzy logic rule based system. It provides an approach that is compatible to the needs of the system. The derivation of probabilities by a statistical model such as HMM provides a solid base for the more syntactical approach of a fuzzy system. HMM yields a more accurate assessment of a log-likelihood which may in turn be used to derive probabilities for the linguistic variables of a fuzzy system. The set of steps employed in the classification phase is given in Fig. 13.

Classification Phase: Steps in Fuzzy Inferencing	
Step 1:	Fuzzification of the input variables: Determine to which degree the input variables belong to each of the appropriate fuzzy sets via membership functions.
Step 2:	Application of the fuzzy operator (AND or OR) to multiple part antecedents: Using as input two or more membership values from fuzzified input variables, a single truth value will emerge as output. The output is the degree of support for the rule.
Step 3:	Application of implication method from the antecedent to the consequent: Using the single value from the antecedent which is the degree of support for the entire rule to shape the output fuzzy set. If the antecedent is only partially true, (i.e., is assigned a value less than 1), then the output fuzzy set is truncated according to the implication method.
Step 4:	Aggregation of the consequents across the rules: Testing is done on all rules in the FIS, and so the rules outputs must be aggregated to make a decision. The inputs to this process are the outputs from each rule's antecedent and the output is one fuzzy set for each output variable.
Step 5:	Defuzzification: The fuzzy set from the aggregated output is defuzzify to produce a single number output from the set.

Fig. 13. Classification Phase: Steps in Fuzzy Inferencing

The classification phase of the system is based on the concept of fuzzy rules developed using Mathlab (Version 7.4.0). The fuzzy inference system (FIS) was built using Mathlab GUI tools from its Fuzzy Logic Toolbox.

A fuzzy rule has two components, an if-part (referred as the antecedent) and a then-part (referred to as the consequent). The structure of fuzzy rule is identical to that of a conventional rule in artificial intelligence. The main difference is the antecedent of a fuzzy rule is described by a linguistic variable and a membership function. A linguistic variable is like a composition of symbolic variable and a numeric variable. Numeric variables are frequently used in science, engineering, mathematics, medicine and many other disciplines while symbolic variables play an important role in AI and decision sciences. Using the notion of the linguistic variable to combine these two kinds of variables into a uniform framework is one of the main reasons that fuzzy logic has been successful in offering intelligent approaches in engineering and many other areas that deal with continuous problem domain.

The features extracted by the HMM yields a very good medium for further conversion of the linguistic variables. The strokes as identified, together with its probability will go through the process of fuzzification. The triangular and the trapezoidal membership functions were used to change the strokes probability into its linguistic variable forms. The two membership functions were chosen due to their simple formulas and computational efficiencies. Examples of the linguistic variables used are : “Very Tall Right_Slant”, “Very Tall Vertical_Line”, as in l’s, or as in some of d’s or b’s. “Tall C_curve” as in c’s, “small C_curve” as appear in d’s, a’s and so on. When the HMM models were used with the fuzzy rules to recognize the handwritten characters, a favorable results was achieved.

8.1 Fuzzy Rule Based

The fuzzy rule-based is comprised of fuzzy if-then rules that are used to formulate the conditional statements that comprise fuzzy logic. A single fuzzy if-then rule assumes the form,

if x is A then y is B,

where A and B are linguistic values defined by fuzzy sets on the ranges (universes of discourse) X and Y, respectively. The if-part of the rule "x is A" is called the antecedent or premise, while the then-part of the rule "y is B" is called the consequent or conclusion. If the antecedent is true to some degree of membership, then the consequent is also true to that same degree.

The antecedent of a rule can have multiple parts, in which case all parts of the antecedent are calculated simultaneously and resolved to a single number using the logical operators as described in the preceding section. Example of such rule as in the rule-based is as below:

If descender is certain and RightCurve is medium and VerticalLine is tall then Char_p is definite

Another example of one the rule from the fuzzy rule-based that describes character ‘d’ is:

If ascender is certain and LeftCurve is medium and Vertical_line is tall then Char_d is definite

The rule above gives a perfect description of character ‘d’. There are other rules that describe all the combination of the linguistic variables that gives the description that would match the character to various degrees. The consequent of a rule can also have multiple parts, in which case all consequents are affected equally by the result of the antecedent. The consequent specifies a fuzzy set be assigned to the output. The implication function then modifies that fuzzy set to the degree specified by the antecedent. The most common ways to modify the output fuzzy set are truncation using the *min* function (where the fuzzy set is chopped off) or scaling using the prod function (where the output fuzzy set is squashed).

The algorithm of fuzzy rule-based inference consists of three basic steps and an additional original step. These steps are : (i) Fuzzy matching: Calculate the degree to which the input data match the condition of the fuzzy rules. (ii) Inference: Calculate the rules conclusion based on its matching degree. (iii) Combination: combine the conclusion inferred by all fuzzy rules into a final conclusion. (iv) Defuzzification: for outputs that need a crisp output and additional step is to convert a fuzzy conclusion into a crisp one. By using the FIS of Mathlab for the purpose mentioned above, the outcome of the inference will be used to determine the classification of the characters.

9. Experimental Results

For all reported results, the following definitions of recognition rate, error rate, rejection rate and reliability rate are used.

Let *B*, be a test set with character images. If the classifier system rejects, N_{rej}, classifies correctly, N_{rec} and misclassifies the rest, N_{err}, then,

$$\text{Recogniton rate} = \frac{N_{rec}}{N_B} \times 100$$

$$\text{Error rate} = \frac{N_{err}}{N_B} \times 100$$

$$\text{Rejection rate} = \frac{N_{rej}}{N_B} \times 100$$

$$\text{Reliability rate} = \frac{\text{Recognition Rate}}{\text{Recognition Rate} + \text{Error Rate}} \times 100$$

The recognition rate, error rate and rejection rate will all summed up to 100%. The calculation of these rates as used by Oliveira (Oliveira et. al., 2002) is felt to be very reflective of the needs of a handwriting recognition system when applied to real applications. The reliability of the system can be demonstrated by the above equation for a given error rate.

The result of the following experiment is based on the recognitions of characters from the IRONOFF database. A sample of not more than 1000 handwritten characters with variability in handwriting is used from that database. Characters chosen from the database are based on its legibility. That means characters that could not be recognize by the naked eye and those written too cursively would be dropped from the test samples. This is because some of the characters in the database are written in a cursive manner and without context it would be quite difficult to be recognized. About 20 to 30 samples of each character are used in the testing. An overall recognition rate of 80.19% is recorded for the system. Table 3 shows all the rates measured for the classification of the lowercase characters in the data set. Table 4 gives the recognition rate grouped by the categories of the characters which is used as one of the distinguishing features in the fuzzy rules.

The features extracted by the HMM yields a very good medium for further conversion of the linguistic variables. The strokes as identified, together with its probability will go through the process of fuzzification. The triangular and the trapezoidal membership functions were used to change the strokes probability into its linguistic variable forms. The two membership functions were chosen due to their simple formulas and computational efficiencies. Examples of the linguistic variables used are : “very tall right slant”, “very tall vertical line”, as in l’s, or as in some of d’s or b’s. “Tall C-curve” as in c’s, “small C-curve” as appear in d’s, a’s and so on.

When the HMM models were used with the fuzzy rules to recognize the handwritten characters, a favorable results was achieved. Bearing in mind that comparing the achievements of various researches in this field is quite difficult as the database and general approaches might differ. Testing done with different databases would give differing results as variations and complexity of the data in the databases are not the same. Similar issues are also with approaches. Approaches would differ in recognition of characters, digits, words, cursive, non-cursive, with post-processing or not. Even though research in the area are extensive many more can be done at not necessarily in improving the percentage of accuracy but also at attempting to reduce complexity of its pre-processing techniques, its classifier, its post-processing and also the need for huge databases.

Data Group	Recognition Rate	Error Rate	Rejection Rate	Reliability Rate
Overall Classification	80.19	8.28	11.53	89.4

Table 3. Recognition, Error, Rejection and Reliability Rate of all characters

Characters grouped in categories of ascender, descender or neither	Recognition rate
Ascenders (e.g. b, d ,f, h, k, l, t)	86.55
Descenders (e.g. g, j, p, q, y)	81.62
Neither (e.g. a, c, e ,i, m, n, o, r, s, u, v, w)	72.40
Average %	80.19

Table 4. Recognition rate of characters in groups

10. Conclusion

The success of the fuzzy rule-based system that is used in recognizing the characters would be quite heavily depended on the accuracy of the features extracted and the way the rules are structured. The work presented by (Lazzerini & Marcelloni, 2000) uses a purely linguistic fuzzy recognizer on handwritten character digit with a recognition rate of 69.5%. Even though it might seem comparatively lower than other methods, the method presented has the novelty in other areas of importance.

With a reasonable rate of recognition on a more difficult database of lower-case characters, HMM model is proven to be a very useful tool to be incorporated into a fuzzy logic rules based system. It provides an approach that is compatible to the needs of the system. The calculation of probabilities for each observation by a statistical model such as HMM provides a solid base for the more syntactical approach of a fuzzy system. HMM yields a more accurate assessment of probabilities for the linguistic variables of a fuzzy system. However the nature of fuzziness in the data captured for the offline handwritten characters recognition research makes a pure statistical approach a little inappropriate. Fuzzy logic has been used in many of the offline researches, giving an impressive result (Wierer & Boston, 2007; Hanmandlu et. al. 2003; Bouslama, 1997). There are many ways of using fuzzy classifier into the problem of handwritten character recognition and this paper proposes a method that does not need huge training sets and is computationally simpler.

The tasks of digit recognition and upper-case character recognition have proven to be simpler than lower-case character recognition. For comparison purposes, some character classifiers recognize some 97% for digits (Lee, 1996; Shouno et. al., 1999), 97% for upper-case letters and 80% for lower-case letters (Heutte, 1998). However, these results are obtained by using complex image processing techniques ((Lee, 1996), or combination feature types, e.g. a combination of structural and statistical features (Heutte, 1998) or complex classifiers (Shouno et. al., 1999). Even though much lower recognition accuracy is achieved by our method comparatively with other methods but on the whole the objective of the research is met. This is to investigate the compatibility of an HMM with a fuzzy rule-based system as the recognizer.

11. References

Atici, A. & Yarman-Vural, F. (2001). "A Heuristic Method for Arabic Character Recognition", *Signal Process*, vol. 62, pp. 87-99.

Arica, N., & Yarman-Vural. F. T. (2001). An Overview of Character Recognition Focused on Off-line Handwriting, *IEEE Trans. On Systems, Man, and Cybernetics*, Vol. 31. No. 2. pp. 216-233.

Blumstein, M. & Verma, B. (1999). A new segementation algorithm for handwritten word recognition, *Proc. of the Int. Joint Conf. on NN, IJCNN '99*, Washington D.C., 878-882.

Bouslama, F. (1997). Arabic character Recognition by Fuzzy Techniques, *in Proc. EUFIT*, Aachen, Germany, pp. 1940-1944.

Bozinovic, R. M. & Srihari, S. N. (1989). Off-Line Cursive Script Word Recognition, *PAM1*, 11, pp. 68-83.

Chen, M. Y.; Kundu, A. & Zhou, J. (1994). Off-Line Handwritten Word Recognition using HMM type Stochastic Network, *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 16, 481-496.

Freeman, H. (1974). *Computer Processing of Line-Drawing Images*, ACM Computing Surveys, 6(1): 57-98 March.

Hanmandlu, M.; Murali-Mohan K. R.; Chakraborty, S.; Goyal, S., & Choudhury, D. R. (2003). Unconstrained Handwritten Character Recognition based on Fuzzy Logic, *Pattern Recognition*, Vol. 36, Issue 3, March, 603-623.

Heutte, L., Paquet, T., Moreau, J.V., Lecourtier, Y., Olivier, C. (1998) A structural/statistical feature based vector for handwritten character recognition, *Pattern Recognition Letters* 19, pp. 629-641.

Impedovo, S.; Ottaviano, L. and Occhinegro, S. (1991). Optical Character Recognition – A Survey, *International Journal of Pattern Recognition & Artificial Intelligence*, 5, 1-24.

Kundu, A. & He, Y. (1991). On optimal order in modeling sequence of letters in words of common language as a Markov chain, *Pattern Recognition*, vol. 24, no. 7, pp. 603-608.

Lazzerini, B. & Marcelloni, F. (2000). A Linguistic Fuzzy Recogniser of Off-line Handwritten Characters, *Pattern Recognition Letters*, 21, pp. 319-327.

Lee, W.L. & Fan, K.C. (2000). Document image preprocessing based on optimal Boolean filters, *Signal Process*, Vol. 80, no. 1, pp. 45-55.

Legault R. & Suen, C.Y. (1997). Optimal local weighted averaging methods in contour smoothing, *IEEE Trans. Pattern Anal. Machine Intell.*, July vol. 18, pp. 690-706.

Leu, J.G. (2000). "Edge Sharpening through ramp width reduction", *Image Vision Computing*, vol. 18, no. 6-7, pp. 501-514.

Madhvanath, S. & Govindaraju, V. (1999). References lines for holistic recognition of handwritten words, *Pattern Recognition,* vol. 32, no. 12, 2021-2028.

Madhvanath, S.; Kim, G. & Govindaraju, V. (1999). Chaincode contour processing for handwritten word recognition, *IEEE Pattern Anal Machine Intell.* , vol. 21, 928-932, September.

Mori, S.; Suen C.Y. & Yamamoto, K. (1992). Historical Overview of OCR Research and Development, *Proceedings of the IEEE*, 80, pp. 1029-1058.

Oh, I.S.; Lee, J.S. & Suen, C.Y., (1999) Analysis of class separation and combination of class-dependent features for handwriting recognition, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, pp. 1089-1094, October.

Oliveira, L. S., Sabourin, R., Bortolozzi, F., & Suen, C. Y., (2002). Automatic Recognition of Handwritten Numerical Strings: A Recognition and Verification Strategy, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 24, No. 11, November.

Plamondon, R. & Srihari, S.N. (2000). On-line and Off-line handwriting Recognition: A Comprehensive Survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, pp. 63-84.

Polesel, A., Ramponi, G., & Mathews, V. (1997). "Adaptive unsharp masking for contrast enhancements", *in Proc. Int. Conf. Image Process*, vol. 1, pp. 267-271.

Rabiner, L.R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, *Proceeding of the IEEE*, Vol. 77, No. 2, February.

Reinhardt, J.M. & Higgins, W.E. (1996). Comparison between the morphological skeleton and morphological shape decomposition, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 951-957, Sept.

Saula, J. & Pietikainen, M. (2000). Adaptive document image binarization, *Pattern Recognition*, vol. 33, no. 2, pp. 225-236.

Serra, J. (1994). "Morphological Filtering: An Overview", *SignalProcess*, vol. 38, no. 1, pp. 3-11.

Shridhar M. & Badreldin, A. (1985) High accuracy syntactic recognition algorithm for handwritten numerals, *IEEE Trans. On Systems, Man, and Cybernetics*, Vol. 15, pp. 152-158, January.

Shouno, H., Fukushima, K., Okada, M. Recognition of handwritten digits in the real world by a neocognitron, in Jain L.C., Lazzerini, B., Knowledge-Based Intelligent Techniques in Character Recognition, CRC Press, Florida, 1999, pp. 19-46.

Solihin, Y. & Leedham C.G. (1999). "Integral ratio: A new class of global thresholding techniques for handwriting images", *IEEE Trans. Pattern Anal. Machine Intell.*, vol.21, August, pp. 761-768.

Sonka, M., Hlavac V. & Boyle, R., (1999). Image Processing, Analysis and Machine Vision, 2nd ed. Pacific Grove CA:Brooks/Cole.

Suen, C. Y. (1986). Character Recognition by Computer and Applications, *in Handbook of Pattern Recognition and Image Processing*, ed. Young T.Y., Fu, K.S., Academic Press Inc., San Diego, CA, pp. 569-586.

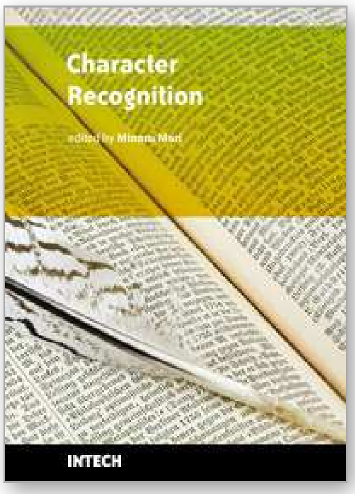
Suen, C. Y.; Tappert, C.C. & Wakahara, T. (1990). The state of the art in on-line handwriting recognition, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 12, pp. 7877-808.

Suliman, A.; Sulaiman, M. N.; Othman, M. & Wirza, R. (2007). Extracting Features for the Lingusitic Variables of Fuzzy Rules Using Hidden Markov Model, *International E-Conference on Computer Science (IeCCS)*, 9-11 July, Greece, AIP Conference Proceedings.

Wierer, J. & Boston, N. (2007). A Handwritten Digit Recognition Algorithm Using Two-Dimensional Hidden Markov Models for Feature Extraction, *IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP*, Vol. 2, April.

Yang, J. & Li, X. B. (1995). Boundary Detection using mathematical morphology, Pattern Recognition Letters, vol. 16, no. 12, pp. 1287-1296.

Zadeh, L. A., (1965). Fuzzy sets, *Information and Control*, Vol. 8, pp. 338-353.



Character Recognition

Edited by Minoru Mori

ISBN 978-953-307-105-3
Hard cover, 188 pages
Publisher Sciyo
Published online 17, August, 2010
Published in print edition August, 2010

Character recognition is one of the pattern recognition technologies that are most widely used in practical applications. This book presents recent advances that are relevant to character recognition, from technical topics such as image processing, feature extraction or classification, to new applications including human-computer interfaces. The goal of this book is to provide a reference source for academic research and for professionals working in the character recognition field.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Azizah Suliman (2010). Hybrid of HMM and Fuzzy Logic for Isolated Handwritten Character Recognition, Character Recognition, Minoru Mori (Ed.), ISBN: 978-953-307-105-3, InTech, Available from: http://www.intechopen.com/books/character-recognition/hybrid-of-hmm-and-fuzzy-logic-for-isolated-handwritten-character-recognition

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821