Arabic Words Recognition by Fuzzy Classifier

Farah Lotfi, Farah Nadir and Bedda Mouldi
Centre Universitaire, Souk Ahras, BP 1553 Souk Ahras 41000, Algeria
Laboratoire d’Automatisme et Signaux, Annaba 23000, Algeria

Abstract: This research study on offline handwriting Arabic words recognition is presented, by the use of a fuzzy proximity measure. In recent years, fuzzy logic has been increasingly used to improve conventional methods especially in pattern recognition fields. The aim of this study was Arabic literal words amount recognition using a fuzzy classifier. We introduce briefly the technique for processing handwritten words, which begins with features extraction, then their classification. The purpose of the classifier was to allocate a class to the test word on a basis of a training set. The fuzzification was introduced in two stages. Firstly to reclassify the obtained K nearest neighbors by a crisp K nearest neighbor approach. Secondly in the classification of the tested word to a class among its K neighbors. The proposed system was tested on 1200 images and a 93.80% classification success rate was obtained.

Key words: Handwritten word recognition, fuzzy nearest neighbors, membership value

INTRODUCTION

Fuzzy logic has been applied to a wide range of applications, including pattern recognition since its introduction in 1965 by Zadeh. Pattern recognition can be viewed as a transformation from Measurement Space to Feature Space, to finally the Decision Space[1]. The patterns processed include 2-D images (like faces, satellite pictures etc.) and 3-D objects for robotic manipulators.

The objective of this study is to develop an Arabic word recognition system. Handwritten word recognition is among the most widely studied fields. It supports not only statistical, structural information and semantic ones, but also some physiological and psychological state of the writer.

These characteristics make handwritten word recognition of consent especially in bank checks area.

Word recognition has become during the later decades almost universal. From that, many automatic systems have been developed and implemented. Most existing systems deal with some constraints such as limited lexicons or restricted writers number, resulting in interesting performances.

However, handwritten recognition deals with variability of scripts and noises generated by acquisition tools (scanner, camera, etc).

To recognize a word, a fuzzy K nearest neighbor[2] is implemented in the Arabic handwritten literal amount recognition system described in this study.

The proposed system we deal with consists of five parts: data acquisition, preprocessing, features extraction, recognition and post classification.

In data acquisition a handwritten literal amount is captured by a scanner, after which preprocessing techniques are used to prepare the image of words for features extraction.

The preprocessing stage begins by dividing the literal amount into words, based on vertical histogram analysis and a computed heuristic (the space between words is about 1.5 times greater than the spaces between sub-words). Then, binarisation is done on the obtained words; this consists of having a bimodal image from gray-levels one[3], then a smoothing is used to filter noises[4].

The third part of our system is features extraction; this part is used to reduce the input vector image by measuring (expressing) it, using certain properties or features of the word image.

The features used by our system are the holistic ones, which are ascenders, descenders, loops, etc. These features are quantitatively extracted from the image and used to recognize words.

For the recognition we use a fuzzy classifier to classify words. After feature extraction we use the vector obtained to compare it against a training set of feature vectors. The classifier tries to match these features to one of the 48 class’s vectors.

The classifier generates a list of candidate words with maximum proximity, which will be used by a syntactic analyzer to make decision about the word which satisfies the grammatical rules designed for this problem.

Corresponding Author: Farah Lotfi, Cites des Peupliers B 9 N 37 Annaba, Algeria Tel: +0021371174736
Ideally the words using the structural features should be well classified. But this is not the case due to the poor features extracted and variability of the script. This is always a certain amount of overlap between classes in the feature space.

In the proposed system a fuzzy nearest neighbor posses advantages of both nearest neighbor and fuzzy systems and are particularly powerful in handling complex, non linear and imprecise problems such as handwritten word recognition. Two membership functions are used, the first one is to reclassify the generated K nearest neighbors and the second one is to classify the tested word according to its K nearest neighbors.

ARABIC WRITING CHARACTERISTICS

The Arabic language has a very rich vocabulary. More than 300 million people speak this language and over 1 billion people use the Arabic language in several religion-related activities. Arabic script is written from right to left. As opposed to Latin one which start from left to right.

The Arabic script is cursive; its alphabet consists of 28 characters. Ten of them have one dot, three have two dots and two have three dots. Dots can be located above, or below. The shape of the character is context sensitive, depending on its location within a word. A letter can have up to four different shapes: isolated, beginning connection from the left, middle connection from the left and right and end connection from the right; example (ملا). Certain character combination form new ligature shapes which are often font dependent. Some ligatures involve vertical staking of characters (محمد). These characteristics complicate the problem of automatically segmenting words into characters (known as analytic approach).

Most of the letters can be connected from both sides; the right and the left. However, there are six letters which can be connected from one side only; the right. These letters are (و), this characteristic implies that each word may contain from one unit or more (sub-words).

Example: (اربعون) (Forty). This word is composed of four sub-words.

FEATURE EXTRACTION

We have been inspired by the human reading process that considers the global high level words shape. For holistic paradigm there is a wide range of methods to word recognition. They can be basically classified in two categories: Statistical and Structural. The statistical method is expressed in terms of partitioning the word feature space. The features are statistics based such as spatial distribution of black pixels, number of black pixels etc.

The structural method is expressed as a composition of structural units and a word is recognized by matching its structural representation with that of a reference words.

The feature extraction step is carried out to determine word’s structures which may be used for recognition. These features are the observables, where the observation provides a value for each of the set of properties.

The main concept is to calculate the number of ascenders, descendents, loops, etc.

Base line detection is the most important information that permits us to situate diacritical dot position and the main part of the word.

The considered vocabulary is composed of 48 words that can be written in an Arabic literal check amount (Table 1).

The boundary following algorithm of the word’s image permits to detect different constituents such as: sub words, loops, ascenders, descendents and diacritical dots.

<table>
<thead>
<tr>
<th>Arabic</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>أربعون</td>
<td>Forty</td>
</tr>
</tbody>
</table>

Table 1: Vocabulary of Arabic literal amounts

The structural features used in our approach are high level ones, which are numbers (Fig. 1) of: descendents, ascenders, loops, one dot above, two dots above, three dots above, one dot below, two dots below. Sub words.

The features extracted are corresponding to 9 structural (Fig. 1) ones according to their possible occurrence numbers in the lexicon’s word:

Three for ascenders, 2 for descendents, 2 for a one dot above, 2 for two dots above, 2 for three dots above, 1 for one dot below, 2 for two dots below, 3 for loops, 4 for sub words.
Example: For the word ٤٥٠٠٠٠ ٤٠٠, we have: 3 ascenders, 1 double dot above, 2 triple dots above, 2 loops, 3 sub words.

**FUZZY K NEAREST NEIGHBOR CLASSIFIER**

(FUZZY K-NN)

The classifier used in our system is a Fuzzy K-NN, which consist on proximity measures. It has been suggested by Pal and Majumder[21].

Fuzzy nearest neighbor classifiers are ideally suited for modeling the non-parametric distribution on handwritten word recognition data.

For the purpose of our system the data were divided as of training and test type. For a given word X, the fuzzy classifier computes the membership X in different classes C₁,…,Cₚ, Cₚ⁺. The membership of X in class Cᵢ can be expressed as μᵢ(X). The test word is allocated to a class for which the membership function yields the maximum value.

After having generated the K nearest neighbors for a test word (by distance similarity), the fuzzification principle is used in two stages. Firstly, it is used in reclassification of the K nearest neighbors obtained by the classical K-NN system. This reclassification tries to redefine class boundaries. Formally we express it by: looking for memberships (by distance calculation) of each neighbor (noted yᵢ) with training classes (noted i class), for every training class we have pᵢ prototypes noted Zᵢ, this membership function is given in (1):

\[
μᵰ(y) = \left[1 + \left(\max_{p = 1...p} \frac{d(yᵰ,Z_p)}{Fd}\right)^{Fe} \right]^{-1} \quad (1)
\]

This function permits to introduce fuzziness, which permits to reclassify yᵰ in classes where it presents the highest membership value. When neighbor’s membership value has been tested with the training set, we compute the membership of test word X noted μᵢ(X) calculated for each of the K nearest neighbor classes, using formula (2):

\[
μᵢ(X) = \left[μᵰ(yᵰ)^* \exp(-a*d(X,yᵰ)/dᵢ) \right] \quad (2)
\]

dᵢ represents the average distance between words of the same class in the reference set.

a, Fe, Fd are constants that determine the degree of fuzziness in membership space, which has been fixed experimentally to the following values: a = 0.45, Fd = 1, Fe = 1. We have used a threshold S that has been fixed to 0, 7.

Since the value of μᵢ(X) increases while the distance value (d) decreases, therefore, for a tested word using a threshold S, a decision rule is stated as follows: let N be the number of classes where the membership function is greater than S, then:

- If N = 0, X is rejected, membership function too low.
- If N = 1 or N>1 and μᵢ(X)is unique, X is recognized.
- If N>1 and μᵢ(X) is not unique, there is ambiguousness.

**SYNTAX BASED POST CLASSIFICATION**

The classification phase has generated a list of candidate words pondered by confidence values, which are the membership values. We will consider from this point that a candidate is a couple of information, the word class and its confidence value.

When obtaining the list of candidate words by the recognition stage, we first sort it by word’s confidence value and then we can consider two cases:

- If there is a word which confidence value is greater than the other and if this word succeeds the syntactic analysis (Table 2), the word is kept and will be part of the resulted literal amount. If on the other hand, the word doesn't satisfy the syntax, it is rejected and the next word of the list will be analyzed.
- If at the head of the list, two words have the same confidence value and satisfy the syntactic analysis, we consider this case like an ambiguity, which can be solved with the use of high level information, the courtesy (numeric) amount for example.

**RESULTS**

For the purpose of the fuzzy K-NN we have constructed four (4) reference sets of different sizes (Table 3), in order to determine the best value of the K parameter and recognition rates.
Table 2: A part of the grammatical rules used

| <Hundred> ::= | <Hund> | <Less_Hund> | <Less_Ten> | +<Less_Hund> | +<Less_Ten> | +<Hund> |
| Hund > ::= | خمسة | ستة | ستة | ستة | ستة | ستة |
| Less_Hund > ::= | <Less_Ten> | <Comp_Nbr> |
| Less_Ten > ::= | نانين | سبع | سبع | سبع | سبع | سبع |

Table 3: Reference sets used

<table>
<thead>
<tr>
<th>Reference set</th>
<th>Set 1</th>
<th>Set 2</th>
<th>Set 3</th>
<th>Set 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of tested words</td>
<td>1200</td>
<td>1200</td>
<td>1200</td>
<td>1200</td>
</tr>
<tr>
<td>Number of words in reference set</td>
<td>96</td>
<td>144</td>
<td>240</td>
<td>480</td>
</tr>
</tbody>
</table>

Table 4: Word recognition rates

<table>
<thead>
<tr>
<th>K</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set 1</td>
<td>85.00</td>
<td>85.00</td>
<td>87.86</td>
<td></td>
</tr>
<tr>
<td>Set 2</td>
<td>91.20</td>
<td>92.10</td>
<td>82.10</td>
<td></td>
</tr>
<tr>
<td>Set 3</td>
<td>92.30</td>
<td>93.10</td>
<td>90.13</td>
<td></td>
</tr>
<tr>
<td>Set 4</td>
<td>92.60</td>
<td>93.80</td>
<td>89.47</td>
<td></td>
</tr>
</tbody>
</table>

We conclude that this performance is very interesting and represents a promising platform, on which more investigations and/or improvements may be done.

REFERENCES


CONCLUSIONS

The Arabic literal amount considered in our case are composed with 48 words. In this study we have used structural features to perform recognition and we have tested our system on a basis of 1 200 words (the 48 words of the lexicon written by 25 different writers). The word shapes are analyzed with a fuzzy classifier obtaining an average recognition rate of 93.80%.