Computing and Informatics, Vol. 24, 2005, 463–479

DECISION FUSION AND CONTEXTUAL INFORMATION FOR ARABIC WORDS RECOGNITION FOR COMPUTING AND INFORMATICS

Nadir Farah, Labiba Souici, Mokhtar Sellami

Laboratoire de recherche en informatique
Université Badji Mokhtar
Annaba, Algérie
e-mail: {farahnadir, souici_labiba}@hotmail.com

Manuscript received 19 March 2004; revised 5 March 2005
Communicated by Pavol Návrat

Abstract. The study of multiple classifier systems has become recently an area of intensive research in pattern recognition. Also in handwriting recognition, systems combining several classifiers have been investigated. An approach for recognizing the legal amount for handwritten Arabic bank check is described in this article. The solution uses multiple information sources to recognize words. The recognition step is preformed with a parallel combination of three kinds of classifiers using holistic word structural features. The classification stage results are first normalized, and the sum combination is performed as a decision fusion scheme, after which a syntactic analyzer makes final decision on the candidate words. Using this approach, the obtained results are very interesting and promising.

Keywords: Arabic word recognition, holistic approach, multiclassifiers, decision fusion, contextual information

1 INTRODUCTION

Several successful methods have been developed to recognize isolated handwritten characters and numerals. Nowadays the research is carried for handwritten word recognition [1, 2, 3], which presents a challenge due to the difficult nature of the unconstrained handwritten words, including the diversity of character patterns, ambiguity of characters, and the overlapping nature of many characters in a word [4].

To achieve better recognition rates for handwritten words, researchers have used different classification algorithms [2]. Therefore, various techniques make different errors and produce different recognition results.

It is very interesting, even if they produce similar results, to envisage the use of these techniques together to benefit from their strength and to take advantage from the fact that mistakes made by them might be different.

There has been considerable research in the last decades on the use of multiple classifier systems for complex classification problems and the potential of performance improvement is proven. Different combination methods are proposed and it is shown that with the use of a set of classifiers providing complementary information for each other the classification accuracy can be highly improved [5, 6]. A multiclassifier system consists of a set of different classification algorithms and a decision fusion for combining outputs.

These last years, a number of papers which analyze the work done on Arabic characters/words recognition have appeared [7, 8, 9]. In this article we are interested in off-line handwritten Arabic words recognition, using a limited lexicon. In this direction some work moved towards Markov models [10], others towards neural models [11] and/or towards the neuro-symbolic systems [12].

The approach is inspired by the human reading process that considers the global word shapes [1] and uses contextual knowledge based on the considered document syntax.

The work leads to the realization of handwritten Arabic literal amount recognition system, based on a global approach, using structural high level features (ascenders, descenders, loops, etc).

The recognition is performed by a multiclassifier system [1]. The proposed system dealt with consists of six parts, namely data acquisition, preprocessing, feature extraction, classification, combination and syntactic analysis.

In data acquisition, handwritten literal amounts are captured by a scanner and then preprocessing techniques are used to prepare the image of words for feature extraction.

The preprocessing stage begins by dividing the literal amount into words, using vertical histogram and a heuristic. Then, binarisation is done on the obtained words; this consists of having a bimodal image from a multigray-level one, then smoothing is used to filter noises.

The third part of our system 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 which are 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. We use three different classifiers for the classification. After feature extraction we provide the vector obtained to the three classifiers which will have different point of view. Each classifier tries to match these features to one of the 48 class's vectors.

A combination module is used on the outcome results by classifiers to compensate for individual classifier weaknesses, and a post classification step permits to validate the combiner propositions.

The remainder of this paper is organized as follows. In Section 2 Arabic writing characteristics are presented, and then a brief overview of the system architecture is done in Section 3. Sections 4 and 5 present preprocessing and features extraction. The three individual classification systems are described in Section 6, and their results in Section 7. A combination approach of classifiers is introduced in Section 8, then the post classification in Section 9. The paper concludes with discussion of the results and an outlook to future work.

2 ARABIC WRITING CHARACTERISTICS

The Arabic language is very rich and difficult by its structure and possibilities. Arabic script is written from right to left. It starts from the right-most position of the page towards the left in a cursive way. The Arabic alphabet consists of 28 basic characters.

The shape of the character is context sensitive, depending on its location within a word. A letter can have four different shapes: isolated, at the beginning, in the middle, at the end. Some Arabic characteristics are particular, we can find for example:

- 10 of them have one dot: ز,ض,ب,ن,ظ,ذ,ف,غ,خ,ج,
- 3 of them have two dots: يـ,تـ,ـقـ,
- 2 have three dots: شـ,ثـ,
- Several characters present loops: و,ة,ـعـ,ق,ف,م,ص ….

The diacritical dots of a character can be located above or below it but not the two simultaneously.

Most of the characters can be connected from both sides, the right and the left one; however, there are six letters that impose a space after (ز,ر,د,ذ,و,ا), they can be connected from the right side only; this is why Arabic language is called semi-cursive. This characteristic implies that each word may be composed of one unit or more (sub-words).

Certain character combinations form new ligature shapes which are often font dependant. Some ligatures involve vertical stacking of characters; this characteristic complicates the problem of segmentation (known as analytic approach) [8, 9].

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

احد	تسعة	ستون	اربعمائة	ألفا	ملياران
اثنان	عشر	سبعون	خمسمائة	الفان	ملايير
ثلاثة	عشرة	ثمانون	ستمائة	مليون	سنتيم
اربعة	اثنا	تسعون	سبعمائة	ملايين	و
خمسة	عشرون	مائة	ثمانمائة	مليونا	دينار
ستة	ثلاثون	مائتا	تسعمائة	مليونان	دنانير
سبعة	اربعون	مائتان	ألف	مليار	سنتيمات
ثمانية	خمسون	ثلاثمائة	الاف	مليارا	جزائري

Table 1. Bank draft lexicon of Arabic literal amounts

3 PROPOSED SYSTEM ARCHITECTURE

The recognition system is constructed around a modular architecture of feature extraction and word classification units.

Preprocessed word image is an input for the structural features extraction module, which transfers the extracted features toward the multiclassifier system (Figure 2). The classification stage is based on three parallel classifiers working on the same set of structural features.

The classifiers results are combined using a statistical decision system. After this combination, a list of candidate words is analyzed by a syntactic module to accept or reject the propositions done by the combiner.

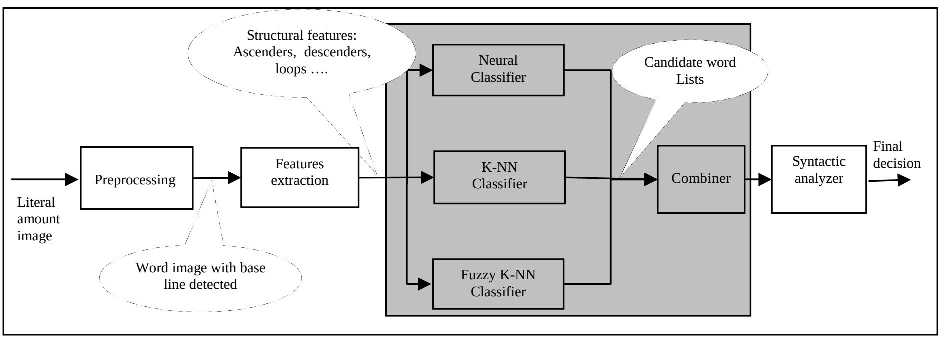

Fig. 1. Global system architecture

4 PREPROCESSING

The word image undergoes a set of processing before extracting the structural features. For the extraction of words from the literal amount, we use a vertical projection method in addition to a heuristic (space between words is 1.5 times greater than the spaces between sub-words).

Binarization. It consists in giving, a bimodal image with two colors (white and black) from a multilevel gray image. We use a thresholding method that determines the threshold from a histogram of the tracing in pixel of the word to recognize. This threshold is calculated on the average of the histogram [14, 15].

Smoothing. It is an operation that permits to decrease noises. In our approach we have been inspired by the algorithm presented in [14]. For a given point P, the algorithm deducts its new value (0 or 1) according to its eight direct neighbors.

Baseline extraction. We adopted the method of baseline detection proposed in [16]. This method consists in doing the horizontal projections of the image and to consider the densest part as being the median one.

5 FEATURES EXTRACTION

The structural features (Figure 2) used in our approach are the structural holistic ones, namely: descenders, ascenders, loops, one dot above, two dots above, three dots above, one dot below, two dots below, sub words.

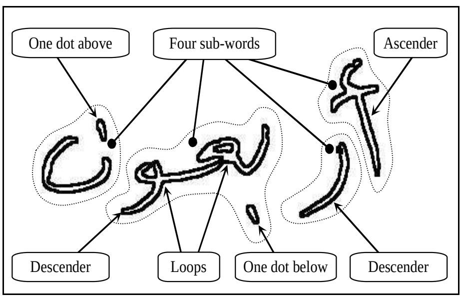

Fig. 2. Word's structural features (forty)

The features used in our system are the global high level ones (holistic) [1]; Table 2 gives the considered lexicon words features.

The feature extraction is based on the image contour. The image contour extraction serves to describe word's image by using Freeman chain codes (Figure 4), representing the image boundaries and topology. The contour encoding is a technique for expressing the digitized boundary of the word image by a sequence of Freeman chain codes specifying the direction in moving from one pixel to another

Arabic words	A	D	ODA	DDA	TDA	ODB	DDB	L	SB	Arabic words	A	D	ODA	DDA	TDA	ODB	DDB	L	SW
خمسة			1	1				2	1	سبعمائة	1			1		1		3	2
ستة				2				1	1	تسعمائة	1			2				3	2
سبعة				1		1		2	1	مائتان	2		1	1				1	3
تسعة				2				2	1	ثلاثمائة	3			1	2			2	3
احد	1								2	ثمانمائة	2		1	1	1			3	3
ثلاثة	2			1	2			1	2	اربعمائة	2	1		1		1		3	4
ثمانية	1		1	1	1		1	2	2	ألف	2		1					1	2
اثنان	2		2		1				3	ألفا	3		1					1	2
اربعة	1	1		1		1		2	3	الفان	3		2					1	3
عشر		1			1				1	الاف	3		1					1	3
اثنا	2		1		1				2	مليون	1	1	1	1				2	2
عشرة		1		1	1			1	2	مليونا	2	1	1				1	2	2
خمسون		1	2					2	2	ملايين	3		1				2	1	2
ستون		1	1	1				1	2	مليونان	2	1	2				1	2	3
سبعون		1	1			1		2	2	ملياران	3	1	1				1	1	4
تسعون		1	1	1				2	2	مليار	2	1					1	1	2
عشرون		2	1		1			1	3	ملايير	2	1					2	1	2
ثلاثون	2	1	1		2			1	3	مليارا	3	1					1	1	3
ثمانون	1	1	2		1			2	3	سنتيم		1	1	2			2	1	1
اربعون	1	2	1			1		2	4	و		1						1	1
مائة	1			1				2	2	سنتيمات	1		1	2			1	1	2
مائتا	2			1				1	2	دينار	1	1	1				1		3
خمسمائة	1		1	1				3	2	دنانير	1	1	2				1		3
ستمائة	1			2				2	2	جزائري	1	2	1			1	1		4

A: Ascender, D: Descender, ODA: One Dot Above, DDA: Double Dot Above,
TDA: Triple Dot Above, ODB: One Dot Below, DDB: Double Dot Above,
L: Loop, SW: Sub-Word.

Fig. 3. Features

on the contour. Figure 4 shows the Freeman chain codes for tracing a contour from a pixel to its neighboring pixels. A contour tracing algorithm based on the algorithm described in [14] is implemented. The algorithm employs the leftmost looking rule, which may be described in terms of an observer walking along pixels belonging to the word and selecting the leftmost pixel available relative to the direction of entry into the current pixel. Scanning of a word starts from top to bottom and right to left in accordance with the characteristics of Arabic language. At every pixel, the neighboring pixels are traced in a sequence dependent on the previous move. The algorithm produces two types of contours, namely external boundary contours (the contours of the main word, sub word, dots) and internal contours (the loops). The external contours are compared to specify the contour of the word main body (which is the largest) and the contours of the dots.

This representation is based upon the work of Freeman [14]. We follow the contour in a clockwise manner and keep track of the directions as we go from one contour pixel to the next.

The codes associated with eight possible directions are the chain codes and, with x as the current contour pixel position, the codes are generally defined like this:

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

5.1 Chain code properties

- Even codes $\{0, 2, 4, 6\}$ correspond to horizontal and vertical directions; odd codes $\{1, 3, 5, 7\}$ correspond to the diagonal directions.
- Each code can be considered as the angular direction, in multiples of 45° that we must move to go from one contour pixel to the next.
- The absolute coordinates $[m, n]$ of the first contour pixel (e.g. top, leftmost) together with the chain code of the contour represent a complete description of the discrete region contour.
- When there is a change between two consecutive chain codes, then the contour has changed its direction. This point is defined as a corner.

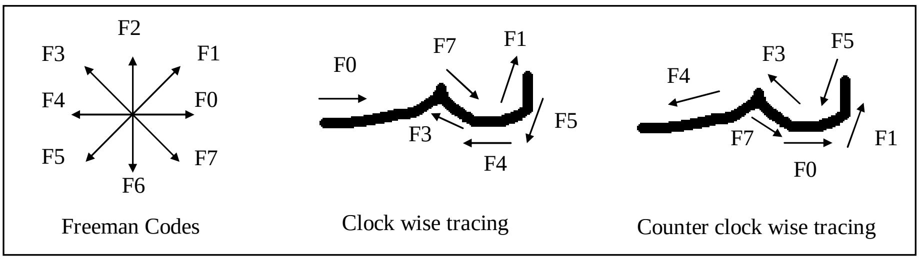

Fig. 4. The eight Freeman codes

To solve the sub-words overlapping problem, a boundary following algorithm has been used inspired by the work done in [14].

For the diacritical dots extraction, we use a heuristic that considers the line thickness as it was done by Ameur et al. [17]. So if S is the line thickness and if we consider the area delimiting components by its coordinates $X_{\min}$, $Y_{\min}$, $X_{\max}$, $Y_{\max}$, we can get the algorithm of 5. The diacritical dot is situated relatively to the baseline.

6 STRUCTURE BASED WORD RECOGNITION

The achieved multiclassifiers system is composed of three different kinds of classifiers, which operate in parallel on the same word's structural features. The three classifiers

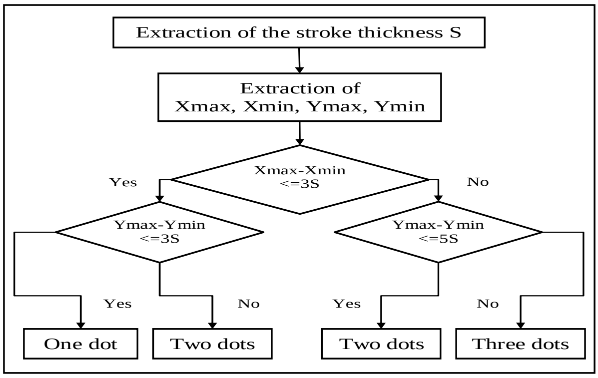

Fig. 5. Diacritical dots extraction algorithm

are: a neural network, a statistical K nearest neighbors system, and a fuzzy K nearest neighbors. A definition of each classifier with their used characteristics will be given in the following subsections.

6.1 The Neural Network Classifier

The used neural network is a multilayered perceptron (Figure 6), with supervised training. The main characteristic of this classifier type is that the classification is based on a training step. The training is materialized by the neurons weights values optimization; this is done with the presentation of representative examples of the considered problem. From these examples, it performs a generalization for new tested words. This generalization ability makes them interesting for classification and recognition problems [18].

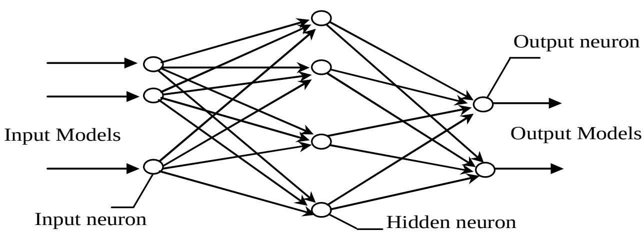

Fig. 6. Artificial Neural Network

This network has a supervised training stage; we give it two different kinds of information: structural features are inputs while the output is a class among the forty eight classes of the lexicon. The training is done by error correction of connection weights with retro-propagation method [19]. Our neural system parameters are:

- An input layer formed by 21 neurons, corresponding to 9 structural features according to their possible occurrence numbers in the lexicon (see Table 1): 3 for ascenders, 2 for descenders, 2 for 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 the number of loops, 4 for sub words number.
- The number of output neurons: 48 neurons (number of lexicon classes).
- The number of hidden neurons: it is calculated by a heuristic: the square root of (input neurons $\times$ output neurons) and then fixed experimentally to 21 neurons.
- The activation function has a sigmoid form.

6.2 The K nearest neighbors classifier (K-NN)

The principle of the K nearest neighbors (K-NN) system consists in searching among the training set (prototype set or reference set), containing the individuals set and their affectation classes, a K number of individuals among the nearest neighbors. We search the nearest neighbors in the sense of distance between feature vectors of the tested word and those of the training set. The closeness of a word to another is typically determined by Euclidean distance. The chosen class will be most represented among the K neighbors.

When the word is tested with the training set, we use a thresholding method to reject or to accept a class. The K-NN classifier conception starts by the creation of the training set, which is constituted of M samples for each of the 48 words of the lexicon, every sample is represented by its 21 features vector. The threshold which permits to reject or to accept the K neighbors under test is the highest value on the representative distance value inter-classes, the representative distance value of a class is the maximal distance value computed between the M samples vectors of the class taken by pair.

6.3 The Fuzzy K nearest neighbors classifier (Fuzzy K-NN)

The fuzzy K-NN uses a method different from that of the crisp K-NN, while the K-NN involves finding the hyper sphere around a word X, which contains K words (independently of their classes), and then assigning X to the class having the largest number of representatives inside the hyper sphere. The fuzzy K-NN method is, however, based on computing the membership of a test word in different classes. We affect the tested word to the class having the highest membership value.

The introduction of imprecision and the uncertainty generated by the fuzzy notion is very well suited to the handwriting recognition problem, since there is influence of the variability of the manuscript, noises generated by operations on the word, the writing style, cause that borders between words classes are overlapping.

We begin by searching the K nearest neighbors of the tested word with a crisp nearest neighbor classifier, then we look for memberships (by distance calculation) of each neighbor (noted Y_j) with training classes (noted i class), for every training class we have p_i prototypes noted Z_p; this membership function [20] is given by (1):

$$\mu_i(y_j) = [1 + (\max d(y_j, Z_p)/F_d)^{F_e}]^{-1} \tag{1}$$

This function permits to introduce fuzziness, which allows reclassifying Y_j in classes where it presents the highest membership value. When membership value has been tested with the training set, we compute the membership of X noted with each of these K nearest neighbors classes, by (2):

$$\mu_i(X) = \{\mu_i(y_j) * exp(-a * d(X, y_j)/d_m)\} \tag{2}$$

d_m represents the average distance between words of the same class in the training set. a, F_e, F_d are constants that determine the degree of fuzziness in membership space, they have been fixed experimentally to the following values: $a = 0,45$, $F_d = 1$, $F_e = 1$. We have used a threshold S that has been fixed to 0.5. Let N be the number of classes where the membership function is greater than S; we have three possible cases:

- if $N = 0$, X is rejected, membership value too low
- if $N = 1$ or $N > 1$ and $\mu_i(X)$ is unique, X is recognized
- if $N > 1$ and $\mu_i(X)$ is not unique, there is an ambiguity.

7 CLASSIFICATION RESULTS

We have used a database containing 4 800 words where 1 200 have been used for training the different classifier. This basis represents the 48 words of the lexicon written by 100 different writers. The test word set includes 3 600 words.

For the neural network, we have obtained a recognition rate of 91 %.

For K-NN and fuzzy K-NN classifiers purpose, we have constructed four training bases (reference bases), in order to determine the optimal value of the K parameter, and to calculate the corresponding recognition rates. While varying the K value and the training bases, we have achieved the results presented in Table 2.

	Recognition Rate					
	K-NN			**Fuzzy K-NN**		
K	1	3	8	1	3	8
Basis 1 (240 words)	82.00	85.00	36.15	85.00	88.00	87.86
Basis 2 (480 words)	86.52	88.40	40.10	91.16	92.16	82.10
Basis 3 (960 words)	88.56	89.08	45.02	92.16	92.16	90.13
Basis 4 (1 200 words)	89.08	89.08	62.00	92.16	92.16	89.47

Table 2. Word's recognition rates for K-NN and fuzzy K-NN

Several remarks are given, concerning these results:

- For $K = 8$, rates lowered distinctly in the K-NN case; it is due to the presence of a majority number of elements from distant classes. The fuzzy K-NN rate, on the other hand, remains stable and elevated enough.
- The parameter K value has been fixed to 3 for these two classifiers, that is the one that gave the best results.
- Recognition rates for the K-NN and the fuzzy K-NN are 89.08 % and 92.16 %, respectively.

We have generalized the K parameter value to the three classifiers. However, its interpretation is different from one classifier to another. For the K-NN, K represents the nearest neighbors; for fuzzy K-NN, K represents the K neighbors with the highest membership value, and for neural system we consider the three most activated output neurons. From this stage we have three words lists where each word is pondered by a confidence value (c.f. 8) granted by the classifier.

Recognition rates for the three classifiers are summarized in Table 3.

Classifiers	Recognition Rates
k-NN	89.09 %
Neural Network	91 %
Fuzzy k-NN	92.16 %

Table 3. Classifiers recognition rates

8 DECISION FUSION

The parallel combination we have adopted is depicted in Figure 7.

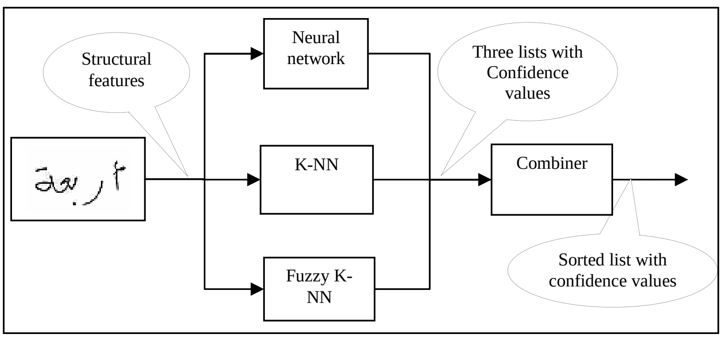

Fig. 7. Parallel classifiers combination

Several strategies are possible to achieve combination: we could add or multiply the confidence values or use maximal/minimal values [21, 22]. Furthermore, all these approaches assume a unique interpretation of the confidence values, for instance as a posteriori probabilities $P(w_i|x)$ for each tested sample x.

For the K-NN classifier, $P(w_i|x)$ is calculated for each class w_i appearing in the word list generated by the classifier [6].

$$P(w_i|x) = \frac{\frac{1}{d(w_i)}}{\sum_{j=1}^{M} \frac{1}{d(w_j)}}$$

where $d(w_i)$ is the distance between the tested word and the class $C_j|_{j=1,\ldots,48}$.

For fuzzy K-NN we have

$$P(w_i|x) = \frac{\mu_i(x)}{\sum_{k=1}^{M} \mu_k(x)}$$

where $\mu_i(x)$ is the membership function (defined in Section 6, formula (2)) for the tested word to class i. For the neural network, each node in the output layer is associated to one class and its output O_i, with [zero to one] range, reflects the response of the network to the corresponding class w_i. To facilitate the decision fusion the responses are normalized and used as estimates of the a posteriori probability of each class [23]:

$$P(w_i|x) = \frac{O_i}{\sum_{k=1}^{M} O_k}$$

After these normalizations we can do combination on the obtained measurements. In our formulation, a confidence transformation method is a scaling function. This latter one rescales the classifiers output to a moderate range (a posteriori probabilities) such that the outputs of the different classifiers are comparable. These transformed confidence measures are desired to approximate the class posterior probability.

In our experiments we used a scheme called score summation [21], each classifier yields as output a list of three candidate words together with their confidence value $P(w_i|x)$. The combination consists in merging the three lists of candidates from the three classifiers to produce a new list by confidence values summing. If a candidate is present in the three lists, its new confidence value is simply the sum of the three previous ones. If a word exists in two lists its confidence value is equal to the sum of the two confidence values. Otherwise its confidence value is equal to the old one. The new list of candidates is re-sorted in decreasing order of confidence values and the candidate at the top of the new list is considered as being the best one. This latter list will be used by the syntactic analyzer to generate a syntactically correct literal amount.

9 SYNTAX-BASED POST CLASSIFICATION

After obtaining a list of candidate words from the classification stage, the combination stage provides three words which will be passed to the syntactic analyzer. The analyzing phase takes into account the previous words accepted to make a decision on the current one.

From a grammar used by the syntactic analyzer (a part is given in Figure 8), the post classification phase makes a decision and generates a winner word from the set of candidates.

```
<literal_Amount> ::=
        <Dinar_Part> +دينار + و +
            <Less_Hund>+ سنتيم |
        <Dinar_Part> +دينار + و +
            <Less_Hund> + سنتيمات |
        <Dinar_Part> +دنانير + و +
            <Less_Hund>+ سنتيم |
        <Dinar_Part> +دنانير + و +
            <Less_Hund> + سنتيمات |
        <Dinar_Part> +دينار + جزائري + و +
            <Less_Hund> + سنتيم |
        <Dinar_Part> +دينار + جزائري +و +
            <Less_Hund> + سنتيمات |
        <Dinar_Part> +دنانير + جزائري + و +
            <Less_Hund>+ سنتيم |
        <Dinar_Part> +دنانير +جزائري + و +
            <Less_Hund>+ سنتيمات |
        <Dinar_Part> +دينار + جزائري |
        <Dinar_Part> +دينار |
        <Dinar_Part> +دنانير +
<Dinar_Part> ::=
        <Thousands> |
        <Thousands> + <Less_Hund> |
        <Thousands> + <Hundreds> |
        <Hundreds> |
        <Less_Hund>
<Thousands> ::=
        ألف |
        و + ألف |
        الفان |
        و + الفان |
        <Less_ten>+ الاف + و |
        <Less_ten> + الاف |
        <Composed_Nbr>+ ألف |
        <Composed_Nbr>+ ألف + و

<Hundreds> ::=
        <Hund>+و +<Less_Hund> |
        <Hund> |
        < Less_ten >+ مائة +و +<Less_Hund> |
        < Less_ten >+ مائة
<Hund > ::=
    خمسمائة | اربعمائة | ثلاثمائة | مائتان
    تسعمائة | ثمانمائة | سبعمائة | ستمائة |
<Less_Hund> ::=
        < Less_ten > |
        <Composed_Nbr>
< Less_ten > ::=
        واحد |
        اثنان |
        <Number>
<Number > ::=
        ستة | خمسة | اربعة | ثلاثة
        تسعة | ثمانية | سبعة |
<Composed_Nbr> ::=
        <Great_Ten>+ عشر |
        < Great_Ten >+ عشر + و |
        < Great_Ten >+عشر+و+<Ten_Nbr>
<Great_ten> ::=
        احدا | اثنا | <Number>
<Ten_Nbr> ::=
        عشرون | ثلاثون | اربعون |
        خمسون | ستون | سبعون | ثمانون
        تسعون |
```

Fig. 8. A part of the grammar on the Arabic checks literal amounts

When obtaining the candidate words list by the combination stage, we distinguish two cases:

- If there is a word whose confidence value is greater than that of the other, and if this word satisfies the syntactic analysis, it is kept, and will make part of the resulted literal amount. On the other hand, if the word does not satisfy the syntax, it is rejected and we analyze the second word in the list.
- If at the head of list we find different words that have the same confidence value, and if the words satisfy the syntax, we retain the one given by the classifier having the greatest recognition rate (see Table 3).

10 RESULTS AND DISCUSSION

Let us note that the recognition rates obtained by the three classifiers (K-NN, Neural network, Fuzzy K-NN) are 89.08 %, 91 %, 92.16 %, respectively (see Section 7). After the decision fusion, the recognition rates reach 94 %. The classification level generates three words lists; decision fusion merges them to generate a single ordered list used by the post classification stage to give a syntactically correct sentence. The list of words can contain up to 9 words; this value is interesting for two reasons:

- It allows the syntactic phase to take a decision on a relatively restricted list; thus, there won't be many inferences.
- It also permits to decrease the probability of rejection by the syntactic phase of candidate words.

An example of mistake that the syntactic analyzer could adjust is given in Figure 9.

In the sentence (see Figure 9) we have five words, during the recognition of these words the first two words that have been proposed by the multiclassifier system have been verified by the analyzer. When arriving at the third word, an ambiguousness concerning the proposition of the multiclassifier (in first proposition) occurs. This word was rejected by the syntactic analyzer, therefore the second proposition of the multiclassifier will first be verified by the analyzer and then accepted or rejected. In the general case, if the three candidate words (after the combination stage) are rejected, the amount cannot be generated. Otherwise, the first word accepted by the analyzer will be kept in the final literal amount.

Correct sentence given by the syntactic analyzer	Sentences proposed by multiclassifiers system
ثلاثة ألاف و خمسون دينار	ثلاثة ألاف **و** خمسون دينار
	ثلاثة ألاف **عشر** خمسون دينار

Fig. 9. Possible mistake example detected by the syntactic analyzer

There are words which have some missing features, or have been ill-detected; in Figure 9 second row, diacritical dots have not been taken into account, and its first character generates a loop. According to the structural shape of the word, we have therefore in the two cases a loop and a descender; they are among the proposed solutions the two words and the analyzer decides according to the grammar.

In the case where the recognition produced candidates having the same confidence values and are given by the same classifier, if the syntactic analysis cannot succeed to a decision, it is a syntactic ambiguity case that can be solved only with higher level information. Among examples of words where the ambiguity remains we have:

- **خمسون**, **سبعون** and **تسعون** and
- **ستة** and **تسعة**

When analyzing the results, we find that the recognition rate is raised to 96 % and the remaining 4 % are owing to:

- 10 %: bad amounts segmentation,
- 20 %: real mistakes in words,
- 30 %: classification mistakes,
- 40 %: absence of handwritten word's feature.

11 CONCLUSION AND PERSPECTIVES

In this work we undertook the recognition of Arabic checks literal handwritten amounts, what implied several processes such as preprocessing, features extraction, classification and post processing. Structural features are extracted from preprocessed word images and presented to three classifiers, which are combined in a parallel scheme. The obtained rate after the combination and post classification stages is 96 %, where the recognition rate has been increased by about 4 % compared to the average classification rate.

These results are interesting and experimentally confirm the assumption that the combination of multiple classifiers decision and the integration of contextual information enhance the overall accuracy of a recognition system.

In fact, the integration of the syntactic analyzer was a very interesting contribution, because the literal check amount does not contain only information on words structure. It is better suited to exploit the logical, lexical, syntactic and semantic relations that exist within the extracted information.

As future perspective for this work, it would be interesting to integrate cooperation with courtesy (numerical) amount recognition. This will permit to solve most cases of ambiguousness signaled in the decision phase. Another way to explore is to offer the possibility of feedback (retroaction) toward the different processing phases (preprocessing, segmentation, recognition, post processing).

REFERENCES

[1] Madhvanath, S.—Govindaraju, V.: The Role of Holistic Paradigms in Handwritten Word Recognition. IEEE Trans. On Pattern Analysis and Machine Intelligence, Vol. 23, February 2001, No. 2.

[2] Steinherz, T.—Rivlin, E.—Intrator, N.: Off-Line Cursive Script Word Recognition: A Survey. International Journal on Document Analysis and Recognition, IJDAR, 1999, Vol. 2, pp. 90–110.

[3] Suen, C. Y.: Reflexions Sur la Reconnaissance de l'Ecriture Cursive. In: Actes CIFED98, 1er Colloque International Francophone sur l'Ecrit et le Document, pp. 1–8, May 1998, Quebec, Canada.

[4] Blumenstein, M.—Verma, B. K.: Neural-Based Solutions for the Segmentation and the Recognition of Difficult Handwritten Words from a Benchmark Database. In: 5th International Conference on Document Analysis and Recognition, pp. 281–284, Banglore, India, 1999.

[5] Zouari, H.—Heutte, L.—Lecourtier, Y.—Alimi, A.: Un Panorama des Méthodes de Combinaison de Classifieurs en Reconnaissance de Formes. In: RFIA 2002, 11th Congres francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle, pp. 499–508, Angers, January 2002.

[6] Xu, L.—Krzyzak, A.—Suen, C. Y.: Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition. IEEE Transactions on Systems. Man, and Cybernetics, Vol. 22, No. 3, May/June 1992.

[7] Amin, A.: Off-Line Arabic Character Recognition: The State of the Art. Pattern Recognition, Vol. 31, 1998, No. 5, pp: 517-530.

[8] Al Badr, B.—Mahmoud, S. A.: Survey and Bibliography of Arabic Optical Text Recognition. Signal Processing, Vol. 41, 1995, pp: 49–77.

[9] Essoukhri Ben Amara: Sur la Problematique et les Orientations en Reconnaissance de l'Écriture Arabe. In CIFED 2002, pp: 1–8, 2002.

[10] Ben Amara, N.—Belaid, A.—Ellouze, N.: Utilisation des Modeles Markoviens en Reconnaissance de l'Ecriture Arabe: État de l'Art. In CIFED '2000, France, 2000, pp. 181–191.

[11] Snoussi-Maddouri, S.—Amiri, H.—Belaid, A.—Choisy, C.: Combination of Local and Global Vision Modeling for Arabic Handwritten Word Recognition. In International Workshop Frontier in HandWriting IWFHR '02, Canada, 2002.

[12] Souici-Meslati, L.—Sellami, M.: Reconnaissance de Montants Littéraux Arabes par une Approche Hybride Neuro-Symbolique. In: RFIA2002, 11th Congres francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle, Angers, Janvier 2002.

[13] Ruta, D.—Gabrys, B.: An Overview of Classifier Fusion Methods. Computing and Information Systems, Vol. 7, No. 1, pp. 1–10, University of Paisley, February 2000. Available on: `http://cis.paisley.ac.uk/ruta-ci0/downloads/paper1.pdf`.

[14] Pavlidis, T.: Algorithms for Graphic and Image Processing. Rockville, MD: Computer Science Press, 1982.

[15] Otsu, N.: A Threshold Selection Method from Gray-Level Histogram. IEEE Transaction Systems, Man and Cybernetics, Vol. 9, 1979, No. 1, pp: 62-66.

[16] Belaid, A.—Belaid, Y.: Reconnaissance des Formes: Methodes et Applications. InterEditions, 1992.

[17] Ameur, A.—Romeo-Pakker, K.—Miled, H.—Cheriet, A.: Approche Globale Pour la Reconnaissance de Mots Manuscrits Arabes. In Actes CNED '94, 3eme Colloque National sur l'Ecrit et le Document, pp. 151–156, Juillet 1994.

[18] Jain, A. K.—Mao, J.—Mohiuddin, K.: Artificial Neural Networks: A Tutorial. IEEE Computer, Special Issue on Neural Computing, March 1996.

[19] Jodouin, J. F.: Les Reseaux de Neurones: Principes et Définitions. Hermes, 1994.

[20] Singh, S.—Amin, A.: Fuzzy Recognition of Chinese Characters. In Proc. Irish Machine Vision and Image Processing Conference (IMVIP '99), Dublin, 8–9 September,

1999.

[21] Duin, R. P. W.: The Combining Classifier: To Train or Not to Train?. In Proceedings 16th International Conference on Pattern Recognition, Quebec City, Canada, August 11–15, 2002.

[22] Jain, A. K.—Duin, R. P. W.—Mao, J.: Statistical Pattern Recognition: A Review. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 1, January 2000.

[23] Kittler, J.—Hatef, M.—Duin, R. P. W.—Matas, J.: On Combining Classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 3, March 1998.

Nadir Farah received his B. Eng. and M. Sc. degrees in computer science from Annaba University, Algeria in 1989 and 1994 respectively. From 1994 until now he is a teacher and a researcher in laboratory of informatics at Annaba University. Currently, he is a Doctor candidate in computer science at the same university. His research interests are pattern recognition, character/word recognition, signal processing and bioinformatics.

Labiba Souici received her B. Eng. and M. Sc. degrees in computer science from Badji Mokhtar University (Annaba, Algeria) in 1992 and 1996, respectively. Since 1996, she was appointed as an Assistant Professor at the Department of Informatics of the same university. Currently, she is a researcher in the LRI laboratory of Badji Mokhtar University and a Ph. D. candidate at the final stage. Her research interests include pattern recognition, Arabic handwritten word recognition, artificial intelligence, hybrid neural systems and perception-oriented systems.

Mokhtar Sellami received his doctorate in computer science from the University of Grenoble (France) in the field of expert systems and logic programming in 1989. He participates in many international research projects in ESPRIT program (European Scientific Project Research in Information and Technology) and in Algeria. He is currently professor, director of LRI laboratory at Badji Mokhtar University and a senior lecturer in artificial intelligence and expert systems. His professional interests include pattern recognition, image processing, information technology and knowledge engineering.