Combination of Local and Global Vision Modelling for Arabic Handwritten Words Recognition

Samia Snoussi-Maddouri, Hamidi Amiri, Abdel Belaïd, Christophe Choisy

RECOGNITION OF HANDWRITTEN WORDS OF LITERAL AMOUNT OF ARABIC CHECKS BY A TRANSPARENT NEURAL NETWORK BASED ON HYBRID FEATURES

S. SNOUSSI MADDOURI, H. AMIRI,

Laboratory of Systems and Signal Processing (LSTS), National Engineering School of Tunis (ENIT), B.P 37 Belvédère 1002, Tunisia

samia.maddouri@enit.rnu.tn, *hamid.amiri@enit.rnu.tn*

A. BELAÏD

Group READ, LORIA-CNRS Campus scientifique B.P. 239 54506 Vandoeuvre-Les-Nancy France

abdel.belaid@loria.fr

We propose in this paper a recognition system of hand-written words of literal amount of Arabic checks based on a specific Transparent Neural Network (TNN). The proposed system tries to simplify the feature extraction step. It uses a combination of a global vision on structural features and a local vision by Fourier Descriptors. The structural feature extraction is realised to be the simplest step. Local normalisation tries to reduce ambiguity of handwriting in comparison to printed script. The whole system is tested on handwritten words extracted from legal amount of Arabic check.

1 Introduction

The computer exceeds the human capacity in performing complex and rapid calculation and providing precise and accurate results. However, it remains limited in several intelligent aspects. Producing automatic recognition script is one of these aspects. Indeed, Human is still more preferment in reading variable handwriting script correctly. The idea of basing an automatic handwriting reader on human reading models has been considered. It has been studied by Taylor in 1983 and various reading models have resulted from this study (such as MacClelland and Rumelhart in perception [6] and Côté in reading of handwriting Latin script [3]).

Several models have been proposed to explain the process of mental lexicon access while reading. They mostly rely on printed Latin texts. Few studies have been interested in the reading of Latin handwriting. No study does take into consideration Arabic handwriting. MacClelland proposes a printed word reading model based on a network, which has a local representation, a parallel processing and an activation mechanism. This model has been applied by Côté to recognize handwritten Latin word. We are interested in applying this idea to handwriting Arabic script. A hybrid model for recognition of cursive Arabic words is proposed in our paper. We here propose a combination of a global vision by Transparent Neural Network (TNN) and a local vision by Fourier Transform (FT). This paper is organized as follow:

Section 2 presents feature extraction. Our first contribution is the generation of primitives specific to Arabic script. Firstly, methods of global feature extraction are described and discussed. Then, we present the contextual segmentation in order to extract the local description of the recognized word. This step would be useful in the retro propagation and normalization step. Section 3 provides a description of the whole TNN recognition system. This section also introduces an amelioration of the TNN architecture so as to adapt it to Arabic word. This is our second contribution. Equally, modification of activation and weight link are justified. A hybridation of TNN by a normalized method is presented in section 4, and constitutes our third and main contribution. The normalization method is based on FT. In section 5, an evaluation of the integration of the TNN recognition system and the FT normalization is done. Conclusion and perspectives are presented in section 6.

2 Feature Extraction

The ultimate goal of our handwritten arabic word recognition system is to imitate the human ability to read at a much faster rate. Psychology studies [6], have proven that the human being read the word globally. He doesn't need to recognize all letters. The structural shape of key letters on its own can be sufficient. If he cannot make it out, he will try to see the word locally. Our recognition system is based on this idea, i.e. combining a local and a global description of the Arabic words.

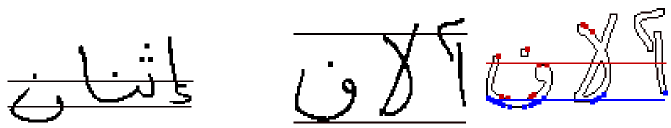
2.1 Global description

The word is considered here as one entity. This method is robust and insensitive to noise and script variation which is the main characteristic of the handwriting. Global features are also easy to be detected. Côté defines two kinds of features: primary features and secondary features. The former takes into account three characteristics: Ascender, descender and loop. The problem with this restricted consideration is that some words, which don't involve one of these characteristics, cannot be processed by her system [3]. By an observation of an Arabic dictionary [1], we can notice that Arabic words contain one of these primitives: "*Ascender*", "*Descender*", "*Loop*", "*diacritic dots"*. The shape of these primitives depends on their position in the word. This position is considered also as primitive and added to one of the four former primitives. When we do not detect any of the described characteristics, we generate "*Nothing*". This primitive is useful to activate words without global feature. Table 1 describes all characteristics taken into consideration by our recognition system.

Table 1. Global Arabic features

Primitives	Description
H	Ascender
J	Descender
B	Loop
P	Diacritic dots above body script
Q	Diacritic dots below body script
R	Nothing of the above primitives
D,M,F,I	Position of the primitives in the word

Baseline detection: A distinguishing feature of the Arabic script is the presence of baseline. The simplest method for images with reduced size such as words and characters is the horizontal projection (figure 1(a)). However this method is sensitive to slant. A first word-pre-processing step is applied in order to eliminate slants [9]. This method is based on FT and will be described in details in section 4. In the case of ambiguity projection method due to the existence of many ascender descender or diacritic points (figure 1(b)), an improvement of the detected baseline is done by the extraction of minimum and maximum (figure 1(c)). In this case, the detection of the base line proceeds by the computing of the local minimum and maximum of the word. Maximum and minimum, which are above lower baseline or below upper baseline deduced from the horizontal projection, are eliminated. From the others we generate new baselines.



(a) (b) (c)

Figure 1. Baseline detection

Ascender and descender extraction: The ascender is defined as a shape having an extension above the upper baseline [4]. The distance between this maximum and the upper baseline shouldn't be greater than certain threshold. This threshold is estimated as (lower baseline – upper baseline)/2. The descender has the same definition for a minimum below lower baseline. Interaction between projection [7], minimum and maximum methods is also used here. Figure 2 presents some of our automatic extraction of ascender and figure 3 presents descender extraction.

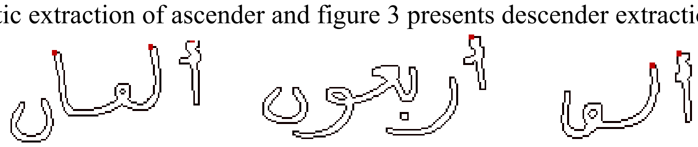


Figure 2. Detection of ascender by Minimum and Maximum method

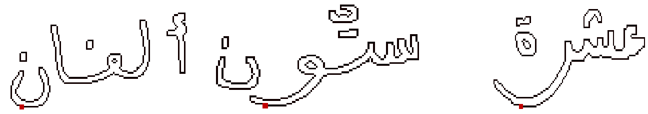


Figure 3. Detection of descender by Minimum and Maximum method

Extraction of the loop: The first idea in loop detection is to use the contour. Going over all extracted contours we detect the closed ones. From these closed contours, we choose those having a number of pixels under a threshold. This threshold is deduced from a statistical study on the recognized words taken from a context (such as literal amount extracted from bank check). It is estimated as 40 pixels. The problem with this method is the confusion possibility with diacritic dots (P, Q, in table 1). Position of closed contours in comparison to upper and lower baseline can distinguish between P, Q, and B. Figure 4 illustrates results obtained.

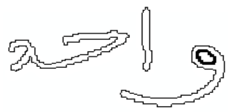


Figure 4. Extraction of loops

Diacritic dot detection: Arabic has 28 basic characters, 15 of which have one to three diacritic dots. Diacritics correspond to a closed contours isolated from the body of the word [1]. They are situated above the upper baseline or below the lower baseline. Their extraction is done during loop detection. The position of these loops in comparison to baseline decides if they correspond to diacritic dot. In Handwriting, dots are generally connected, that is why the number of diacritics is not taken in to consideration here. Figure 5 presents some results.

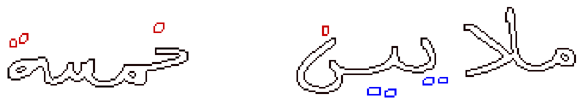


Figure 5. Diacritic dot detection

Position of primitives (characters): The shape of an Arabic character depends on its position in the word. A character might have up to four different shapes depending on being isolated, in the beginning, at the end, or in the middle of a word [1]. This position is detected during the feature extraction. Indeed, extracted zone are delimitated by local minimum. The minimum comes from the vertical projection and the contour (figure 6). The number of black pixels is computed in the neighbourhood of extracted zones boundary and between upper and lower baseline. If this number is >0 in the left boundary and =0 in right side, the position of the characteristics is in the beginning (figure 4).

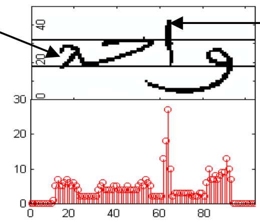


Figure 6. Detection of primitive positions

Feature extraction is the most difficult and important problem of script recognition. Our own is to simplify this step. We have applied simple and classic extraction methods. An evaluation before its integration in the TNN is done and a rate of extraction of each feature is presented in table 2. Error rate would be eliminated by the insertion information during the retro propagation and normalization steps.

Table 2. Evaluation results of feature extraction

Features kind	Number of visual Primitives	Number of automatic Detected primitives	Extraction rate
Ascender: H	360	295	81.94%
Descender:J	323	245	75.85%
Loops : B	356	322	90.45%
Diacritics:P	1134	1062	93.65%
Diacritics:Q	321	258	80.37%

2.2 Local description

Local description is necessary when the character hasn't got any of the above primary features, or if these features are wrongly detected (such us the open loop). However, the main problem with local processing is pre-segmentation step [2] [1]. During the matching between a zone in the image with a letter in a word of the lexicon, Côté considers that all letters have the same width, that it varies according to the number of letters in a given word and that it is known a priori.

This consideration cannot stand when it comes to the Arabic script. Indeed, neither the printed nor handwritten Arabic characters have a fixed size (height and width). Our goal is not to segment the whole word into letters, but to use zones already detected in global feature extraction step to locate positions and delimitate unknown zones. As a matter of fact, in order to reduce the segmentation problem, we start by extracting only segments between two already delimitated primary features. Then, we try to extract the first character and the last one. In this way, we have only one difficult position to detect (figure 6). The first simple idea is the vertical projection [2]. The delimiting point comes from the minimum of the histogram [7]. The projection method is simple and gives a first estimation of segment boundaries, but it fails in the cases of overlapping or scripts with a great slant. The second idea is to detect minimum from the contour [1]. Only minimums belonging to the neighbourhood of already extracted zones, delimitated by projection method, are maintained. The extracted contour is copied in other image file (figure 7). This image is the in put of the local processing step.

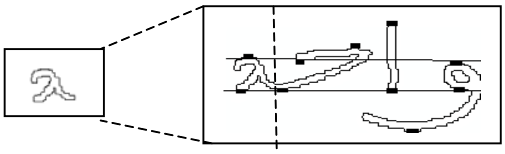


Figure 7. Contextual pre-segmentation

3 Transparent Neural Network (TNN)

Networks with local representation are interesting when the amount of data to be represented is small and when the data can be described with simple relations. The behaviour of this type of network can be explained step by step. Thus, it falls into category of “transparent” systems [3]. For this reason, we call it Transparent Neural Network (TNN). The particularity of this system is the transparent and parallel processing of information and the progressive propagation of activation between adjacent cells. Its advantage is that it doesn’t require a learning stage or large databases.

3.1 Architecture of TNN

The system developed by Côté and called PERCEPTRO, is composed of three layers: features layer, letters layer and words layer. An Arabic word is composed of one or more connected components [1]. We will refer to each connected sub-word as Pieces of Arabic Word (PAW) (see layer three, four, figure 8). Contrary to PERCEPTRO, we introduce a fourth cells layer between letters and words ones, which describes PAW’s layer. Figure 8 illustrates our TNN architecture. We have a cell for each word in the pre-defined lexicon of literal amount of Arabic checks, as well as for each PAWs of these words, each letter and each feature associated with a given location in the image. There are 71 words, 62 PAWs, 61 letters and 10 features. In each of neural cells we found only structural description of letters, PAW’s and words and not corresponding images that is why it isn’t neither time nor memory consuming.

Cell activation: Every cell in each cycle (*c*) has activation. In the beginning all cells are initialized to *0*. The cells activation depends on their actual activation and on that of their neighbours according to the following equation:

$$A_i(c + 1) = (1 - \theta)\, A_i(c) + E_i(c) \quad (1)$$

$A_i(c + 1)$ being the activation of cell i at the cycle $c + 1$, θ is a constant for the unit decay set to *0.07* by Maclelland [3], and $E_i(c)$ is the effect of the cell neighbours i. This effect is defined as: $E_i(c) = n_i(c)(1 - A_i(c))$ (2)

$n_i(c)$ being the excitation of the neighbours of the cells i, it is defined as :

$$n_i(c) = \sum_{j=1}^{nn} \alpha_{ij} A_j(c) \qquad (3)$$

nn is the number of neighbours j of the cell i and α_{ij} is the weight of the connection between i and j. Examples of activation values are presented in figure 8.

Connection weights: Five kinds of weights are defined according to the degree of influence of the neighbour j on the cell i:

- $\alpha_{FL} = 1/NF$: weight between Features and Letters layer. *NF* is the Number of Features extracted from word image in the same zone. This adds to the contribution of each features on the activation of the corresponding letters. In figure 8 we have two and three extracted features so $\alpha_{FL} = 1/2$ or $1/3$.
- $\alpha_{LP} = 1/NL$: weight between Letters and PAWs layer. *NL* is the Number of PAW Letters. This can give the same degree of activation to all the letters of the PAW. For instance, the weight link to the tenth cell in the third layer in figure 8 is *1/4* because the corresponding PAW is composed of four letters. Here Côté introduces a fuzzy function to estimate the position of the letter in the word. We see that only structural description of the position, as "*D, M, F, and I*" is sufficient to activate PAWs.
- $\alpha_{PW} = 1/NP$: weight between PAWs and Words layer. *NP* is the Number of PAWs in the Word.
- $\alpha_{WP} = 1/NW$: weight between Words and PAWs layer. *NW* is the Number of Words, which contain the PAW. All these Words have been activated by this PAW and would contribute to its validation.
- $\alpha_{PL}=1/NP$: weight between PAWs and Letters layer. *NP* is the Number of PAWs activated by a letter.

TNN tries to recognize a word from its visual structural features by the use of two processes. A Bottom-up or propagation process, and a top-down or retro propagation process:

3.2 Bottom-up process

It propagates information from the handwritten image to one of word lexicon. Each zone of the handwritten image can be described by more than one feature (see zone 3 in figure 6). A letter is activated by the whole set of features of a given zone. Activated letters contribute to the activation of PAWs. The position of extracted zone is taken into account in this step. Before the activation of words, a first step of validation of activated PAWs is done. Only PAWs activated by a number of letters equal to the number of zones in the same image PAWs are held for the following steps of recognition. Other PAWs are cancelled. Indeed, if we extract loop in the beginning and descender at the end, only PAWs having these two characteristics in the different position are maintained. If there is some PAWs activated by only one of these two characteristic zones, they should be cancelled. The words concerned with the held PAWs are also activated. The order of PAWs in the word is easy to be detected and is equally taken into consideration during the activation step. Figure 8 illustrates this process with all the activation values.

3.3 Top-down process

This step start by the validation of activated PAWs which validates at their turn activated letters. This validation increases the corresponding activation value. In figure 6 we can see an improvement of the activation of the cell number eight in the third layer. Activated PAWs generate letters hypotheses, which give some hints about the identity of the unknown letters present in the image [Côté 97] [7]. Cells number eight, eleven, twelve and thirteen in the third layer, propose three letters. These hypotheses are checked against the real image by the use of Fourier Transform normalization and decision.

3.4 Decision

The succession of bottom-up and top-down processes constitutes a cycle. The proposed letters constitute the input to the next cycle. The output is a list of activated words taken from a printed lexicon. The word recognized is that of the highest activation (see figure 6). When we have the same activation for more then one word, a FT normalization step is applied on letters with primary features. This step is very useful for short words. Contrary to Côté we can have a decision after only two cycles.

4 Fourier Transform Normalization (FTN)

The FTN step integrated in retro propagation step is our major intervention on TNN. Its advantage is to resolve the problem of secondary features, which are difficult to describe and to detect. In fact, from a set of proposed characters, Côté [3] tries to verify the existence of secondary features such as position of loops, a “t” shape and bottom-up and top-down valley. These features are more difficult to be extracted especially with the variability of handwriting.

In their study of handwriting, some authors are interested in the variability of the handwritten script in comparison to the printed one. Some of them have thought of cursive normalization [8]. The objective of our work is to create a relationship of similarity between a handwritten character taken from a word context and a reference set of characters proposed by the propagation step of the word recognition system. These letters are considered as a reference for the normalization step. So, the principal objective of normalization is to help the TNN in choosing one of the characters proposed by the top-down process. Bringing the local unknown form near one of the proposed characters does this.

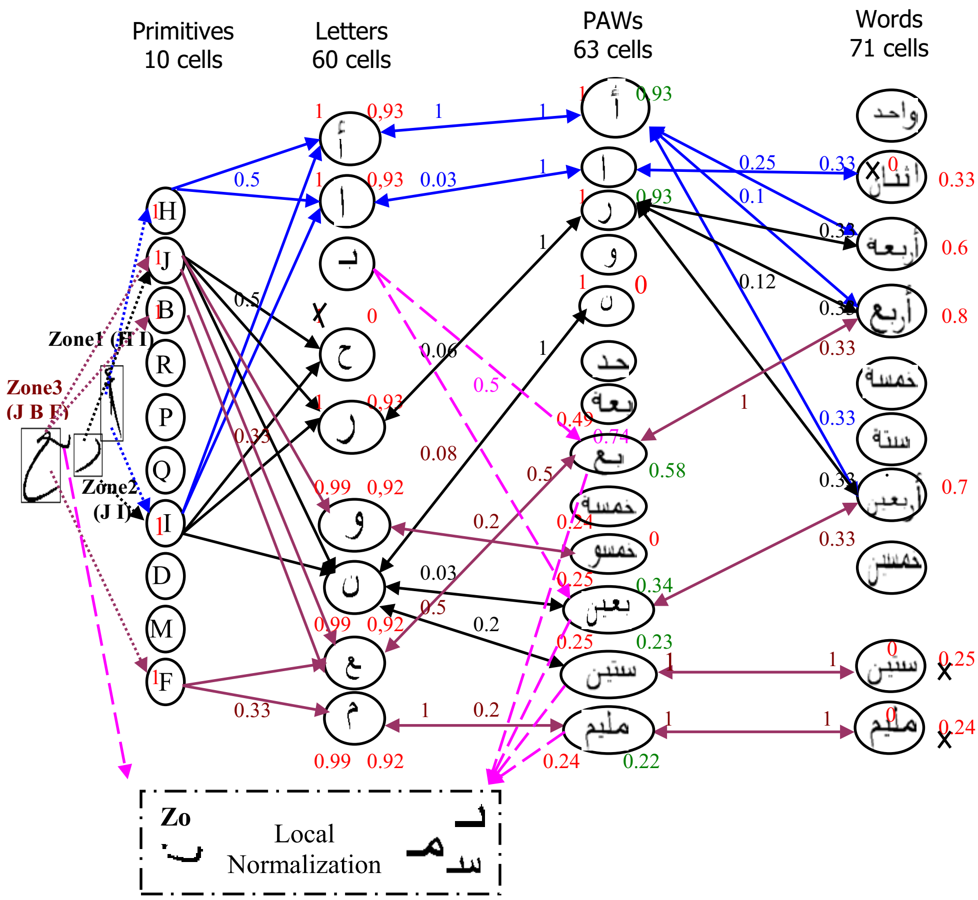


Figure 8. Architecture of TNN for Arabic word recognition

4.1 Normalization procedure

Our normalization approach uses the boundary function of the characters pre-segmented from the image. It is based on the method developed by Kuhl [5]. Normalization is composed of four steps. In the first step, we start by boundary detection [8]. In the second step, the Freeman chain code is generated. A Fast Fourier Transform (FFT) algorithm calculates the Fourier coefficients (a_n, b_n, c_n, d_n, A_0 and C_0) of a chain-coded contour defined in equation (4), with N is the number of boundary points:

$$Y_N(k) = C_0 + \sum_{n=1}^{N} c_n \cos\frac{2n\pi k}{N} + d_n \sin\frac{2n\pi k}{N} \quad (4)$$

$$X_N(k) = A_0 + \sum_{n=1}^{N} a_n \cos\frac{2n\pi k}{N} + b_n \sin\frac{2n\pi k}{N} \quad (5)$$

The forth step attempts to normalize these coefficients in order to cope with orientation and size variation of a handwritten character. This normalization is based on the elliptic loci of the projections points (Xn(k), Yn(k)) (see figure 9).

Original Image 1st harmonic 1st + 2nd harmonic 1st + 2nd +10th harmonic

Figure 9. Fourier approximation, of the boundary of the character "ba ".

Position normalization: To make Fourier coefficients of our handwritten character independent of translation, we can ignore the bias terms A_0 and C_0.
Starting point transform: In order to obtain the same elliptic loci for different starting points, a step of normalization is necessary. Indeed, a difference between the starting points is displayed in the projected space as a phase shift. The starting point angular rotation θ_1 is determined from the point (x1, y1) with elliptic loci by:

$$x_1 = a_1\cos\theta + b_1\sin\theta$$
$$y_1 = c_1\cos\theta + d_1\sin\theta \quad \text{and} \quad \theta = 2\pi k/N \quad (6)$$

The magnitude of the first harmonic phase is:

$$E = \sqrt{(x_1^2 + y_1^2)} \quad (7)$$

By differentiating the magnitude of the first harmonic phase and setting the derivative equal to zero, we obtain :

$$\theta_1 = \frac{1}{2}\arctan\left(\frac{2(a_1 b_1 + c_1 d_1)}{a_1^2 + c_1^2 - b_1^2 - d_1^2}\right) \quad (8)$$

This expression locates the first semi-major axis to occur moving away from the starting point in the direction of the rotation around the contour [8].

Harmonic phase normalization: As it is shown in figure 7, first harmonic phase describe the global shape of the character. The other harmonics add details in order to approximate the original contour. So, the most useful harmonic for normalization is the first one. . The idea of normalization is to rotate the first harmonic phase until it is aligned with the semi major axis of its loci. Then, the coordinate axes U, V defined by the major and minor axis of the ellipse associated to the first harmonic are rotated into the X, Y coordinate axes in which the contour was originally oriented (figure 10). The spatial rotation is determined from the Fourier coefficients a_1* and c_1* that are corrected for starting point displaced θ_1 (see equation 9).

$$
\begin{aligned}
x_1^*(t^*_k) &= a_1^* \cos \frac{2\pi}{N} k^* + b_1^* \sin \frac{2\pi}{N} k^* \\
y_1^*(t^*_k) &= c_1^* \cos \frac{2\pi}{N} k^* + d_1^* \sin \frac{2\pi}{-N} k^*
\end{aligned}
\qquad (9)
$$

The harmonic phase ψ_1 is obtained as:

$$
\psi_1 = \arctan\left(\frac{y_1^*(0)}{x_1^*(0)}\right) = \arctan\frac{c_1^*}{a_1^*} \qquad (10)
$$

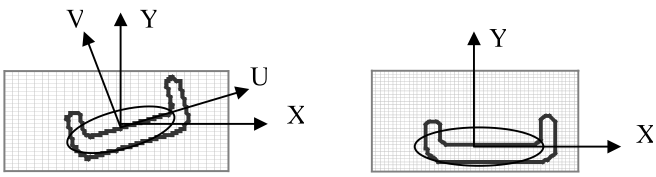


Figure 10. Harmonic phase normalization

Size normalization: The normalization of the size can be made by dividing each of the coefficients by the magnitude of the semi major axis defined as :

$$
E^*(0) = \sqrt{(x_1^*(0))^2 + (y_1^*(0))^2} = \sqrt{a_1^{*2} + c_1^{*2}} \qquad (11)
$$

This transformation proposed in [8] tries to reduce the first harmonic size of all characters to 1.

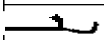
4.2 4.2 Insertion

In order to decide about the insertion of a new letter in the next propagation step, a metric distance D(ma,Im), between images of handwritten characters and their printed references is defined as follow:

$$D(ma,Im)=\frac{\sum_{i=1}^{N}(a_{ma}^2-a_{Im}^2)+(b_{ma}^2-b_{Im}^2)+(c_{ma}^2-c_{Im}^2)+(d_{ma}^2-d_{Im}^2)}{\sum_{i=1}^{N}(a_{ma}^2+a_{Im}^2+b_{ma}^2+b_{Im}^2+c_{ma}^2+c_{Im}^2+d_{ma}^2+d_{Im}^2)} \qquad (12)$$

Table 3 presents distances between proposed letters and unknown ones in the example of figure 6 before and after normalization. These distance help us decide about the nearest printed character to be inserted in the following cycle.

Table 3. Distances between unknown zone and proposed letters before and after normalization

5 Unknown zones	6 Proposed letters	**Distances Before Normalization**	**Distances After Normalization**
		0,73	0,05
		1,00	1,08
		0,65	0,49

5 Experiments and Results

Experiments are done on the 63 PAWs and then on the on the 69 words. For PAWs experimentation only three layers are maintained. The evaluation is done in four steps. Firstly, we evaluate the TNN without integration either of feature extraction or of FT normalization. The features are visually described by the person and introduced to the system. Secondly, we integrate the automatic feature extraction with recognition TNN and try to see its impact on the recognition rate. These experimentation are applied to handwritten words extracted from legal amount of Arabic check multiscriptors. 2070 images are used as 30 samples for each of the 69 words. The third experiment concerns the improvement of TNN by FT without feature extraction and then with automatic feature extraction. We apply the same experimentation on the 2070 word images. Results are given in table4.

We observe that the rate of recognition decreases after integrating of feature extraction. So, recognition rate depends on the feature extraction quality. The error rate before automatic feature integration is due to some PAWs, which have the same global description. Fourier descriptors are able to describe details. They can eliminate ambiguities and improve recognition rate. However error rate after FT integration is due to the fact that we don't take into account the number of diacritic dots. Word recognition rate is greater than PAWs rate because we have more information such as the number of PAWs and their position in the words.

Table 4. Recognition rate for all words of literal Arabic amount

Combination possibilities	PAWs Recognition rate	Words Recognition rate
TNN and manual feature extraction	84,21%	98%
TNN + FT and manual feature extraction	97,36%	100%
TNN and automatic feature extraction	68,42%	90%
TNN + FT and automatic feature extraction	95%	97%

6 Conclusion and perspectives

We have present in this paper the major contributions of our system on Arabic word recognition. The choice of characteristics, the TNN architecture specific to Arabic and the hybridation by FT is justified. Indeed, our TNN system introduces characteristics specific to the Arabic script. Four letter positions in the word give more information about word characteristics. The characteristic "*R: nothing*" guarantees the processing of all Arabic words. The normalization by FT resolves the issue of secondary feature and improves the step of information insertion in order to move to propagation/retro propagation cycle. Each step of feature extraction, segmentation, and normalization is experimented before their integration to the TNN recognition system. This experimentation results are estimated to 79% for feature extraction rate (table 2), and 98,51% for rate separation between extracted zone and proposed letters after FT normalization [8]. The segmentation efficiency is evaluated as its impact on FT separation rate. The recognition rate of PERCEPTRO was up to 92%. The improved TNN rate is estimated to 97%.

As our first perspective we work on the improvement of feature extraction. The construction of a professional database is our second perspective; it aims at testing on a wider variety of scripts. Samples used for experimentation are first results of our LSTS database. This database would be an experimentation reference for all Arabic recognition systems and constitutes our second perspective. The problem with local processing is the segmentation. Despite the contextual information given by the TNN retro propagation step, delimitation of zones necessitates more efficient method. Hidden Markov Model segmentation method is our third perspectives. The new TNN architecture proposed in this paper try to recognize Arabic word. However, it is easier to extract PAWs then words. Our last perspective is to use TNN for the recognition of the whole literal amount. So, amount layer can replace word layer. This can prepare us to a possible interaction between TNN system for literal amount recognition and a digital amount recognition system.

7 References

1. B.Al-Badr, S.A.Mahmoud, "Survey and bibliography of Arabic optical text recognition", Signal processing, pp:49-77, 1994.
2. R.G.Casey, E.Lecolinet, "A Survey of Methods and Strategies in Character Segmentation", IEEE, Transaction on Pattern Analysis and Machine Intelligence, vol 18, n°7, 1996.
3. M.Côté, M.Lecolinet, E.Cheriet, M.Suen, " Automatic Reading of Cursive Scripts Using a Reading Model and Perceptual Concepts", IJDAR, Vol1, pp:3-7, 1998.
4. D.Guillevic, "Unconstrained Handwriting Recognition Applied to the Processing of Bank Checks", a thesis in the department of computer science, 1995.
5. F.P.Kuhl, "Elliptic Fourier Features of a Closed Contour", Computer Graphics and Image processing, vol 18, pp:336-258, 1982.
6. JL.McClelland, DE.Rumelhart, " An interactive activation model of context effects in letter perception", Psychological Review 88, pp: 375-407, 1981.
7. S.Snoussi Maddouri, H.Amiri, "Une Méthode de Reconnaissance de Mots Manuscrits Arabes par Réseaux de Neurones Transparent", Conférence Internationale en Informatique, Algerie, 1999.
8. S.Snoussi.Maddouri, H.Amiri, A.Belaid, "Local Normalization Towards Global Recognition of Arabic Handwritten Script", 4th IAPR International Workshop on Document Analysis Systems, Brazil, 2000.
9. S.Snoussi.Maddouri, V.MÄrgern , H.Amiri, "Normalization of Handwriting Pieces of Arabic Word : two different approaches", Tunisian-German Conference Smart Systems and Devices, Tunisia, 2001.