American Journal of Intelligent Systems 2014, 4(2): 32-42
DOI: 10.5923/j.ajis.20140402.02

Character Recognition by Frequency Analysis and Artificial Neural Networks

Hajji Tarik[*], **Jaara El Miloud**

Laboratoire de recherche en informatiques LRI, Faculté des sciences FSO, Oujda, 6000, Maroc (LaRI/FSO/UMP1)

Abstract This paper describes an artificial neural network (ANN)-based system that uses a word frequency database for optical character recognition (OCR) of words in Amazigh - a North African language. OCR is a widely researched field with commercial systems available for common languages like English. In this research, we will study the character recognition using a new approach based on statistical calculation and artificial neural networks. What we propose in this work is a method that will exploit the power of the semantic analysis of the language to guess all the characters in order to solve the problem of similarity of characters and reduce recognition time. The approach we propose in this study will increase the recognition rate and optimize the response time. We have already several methods based on artificial neural networks, but most of these methods don't make an exact recognition with a rate of 100%, also the response time for these methods is very important.

Keywords Artificial Neural Networks, Database, Modeling, Probability and Statistics, Natural Language Processing, Image Processing

1. Introduction

Optical character recognition (OCR) or video-coding means computer processes for translating text images printed or typed text files. It achieves far less than the human being, who, in addition to the recognition, understands the message, stores it or starts a critical analysis in a single time. And it's a very important topic area in the image processing science and automatic language processing.

Several approaches and methods are invented to solve the problem of characters recognition, but the common problem among the majority of these approaches is response time, which is relatively important depending on the size of the instance and the recognition rate being usually lower than 100%. This work aims to find a quadratic algorithmic approach for the optimization of the temporary complexity of the Amazigh language recognition process, and also to maximize the rate of recognition of these characters facing some recognition problems. Many recognition methods have been proposed most of which is based on the treatment of individual character such as the extraction of the characteristics of each character. This type of method is very important although it does not meet 100% we need, because we still have characters not properly classified or unclassified. In the software world, we have a large collection of computer solutions in this area as: More Data, More data Fast, and GOCR. The purpose of this paper is a method that will use the power of the semantic analysis of the language to guess all the characters in order to solve the problem of similarity of characters and reduce recognition time.

The Amazigh language is one of the ancient languages of the world (950 years before Christ) and is used in Africa (Algeria, Canaries, Egypt, Libya, Mali, Mauritania, Morocco, Niger, and Tunisia). Being recognized as an official language in North Africa, it is still in the early stages of computerization.

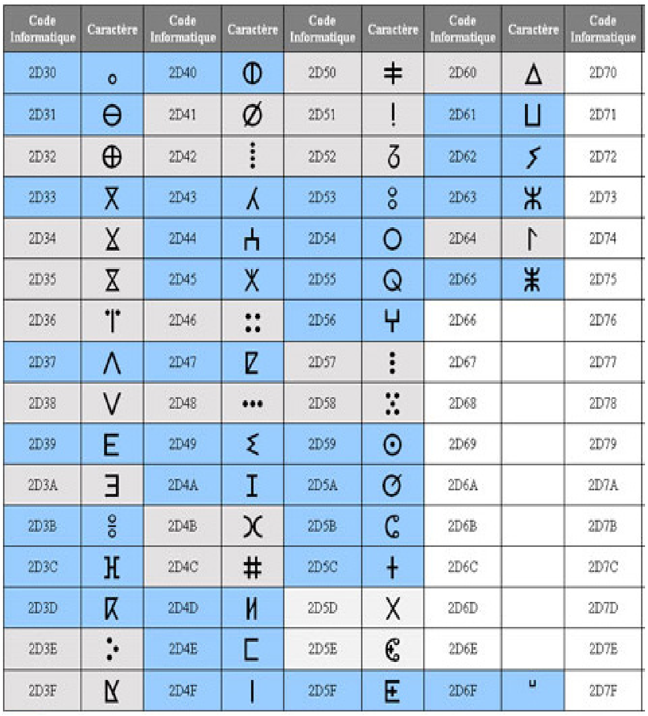
1.1. Tifinagh Alphabet

Before the standardization of the Amazigh, the writing as well as the computer modelling was based on the Arabic alphabet or the Latin alphabet, enriched by the special characters of the International Phonetic Alphabet (IPA). Currently, the Language Planning Center (CAL) recommends the Tifinagh as the alphabet characters for the Amazigh language.

From version 4.1.0 of the Unicode standard, the IRCAM characters are encoded in the range U + 2D30 to U + 2D7F. There are 55 characters defined in the standard, but there are many more characters that are not part of the Unicode standard [1]:

[*] Corresponding author:
hajji-tarik@hotmail.com (Hajji Tarik)

Published online at http://journal.sapub.org/ajis



Code Informatique	Caractère	Code Informatique	Caractère	Code Informatique	Caractère	Code Informatique	Caractère	Code Informatique
2D30	ⴰ	2D40	ⵀ	2D50	ⵐ	2D60	ⵠ	2D70
2D31	ⴱ	2D41	ⵁ	2D51	ⵑ	2D61	ⵡ	2D71
2D32	ⴲ	2D42	ⵂ	2D52	ⵒ	2D62	ⵢ	2D72
2D33	ⴳ	2D43	ⵃ	2D53	ⵓ	2D63	ⵣ	2D73
2D34	ⴴ	2D44	ⵄ	2D54	ⵔ	2D64	ⵤ	2D74
2D35	ⴵ	2D45	ⵅ	2D55	ⵕ	2D65	ⵥ	2D75
2D36	ⴶ	2D46	ⵆ	2D56	ⵖ	2D66		2D76
2D37	ⴷ	2D47	ⵇ	2D57	ⵗ	2D67		2D77
2D38	ⴸ	2D48	ⵈ	2D58	ⵘ	2D68		2D78
2D39	ⴹ	2D49	ⵉ	2D59	ⵙ	2D69		2D79
2D3A	ⴺ	2D4A	ⵊ	2D5A	ⵚ	2D6A		2D7A
2D3B	ⴻ	2D4B	ⵋ	2D5B	ⵛ	2D6B		2D7B
2D3C	ⴼ	2D4C	ⵌ	2D5C	ⵜ	2D6C		2D7C
2D3D	ⴽ	2D4D	ⵍ	2D5D	ⵝ	2D6D		2D7D
2D3E	ⴾ	2D4E	ⵎ	2D5E	ⵞ	2D6E		2D7E
2D3F	ⴿ	2D4F	ⵏ	2D5F	ⵟ	2D6F	ⵯ	2D7F

Figure 1. Tifinagh characters

This alphabet includes: 27 consonant (lips, the dental, the alveolar, the palatal, the Velar, the labiovelars, the uvular, the pharyngeal, the laryngeal), the semi consonants and the vowels.

1.2. Tamazight (state-of-the-art)

Amazigh is an Aboriginal language of Northern Africa. This language brings together thirty dialectal varieties present from the Mediterranean to the South of Niger, and form the Canaries Islands to the western borders of Egypt. In Morocco, Amazigh is divided into three large regional varieties covering the whole mountainous regions: the Tarifite to the Northeast; Tamazight in the Middle Atlas and a party of the High Atlas; and Tachelhit in the South and Southwest. The Amazigh language and culture has experienced, through many steps, a process of standardization and integration in the new technologies of information and communication: specific encoding by extended ASCII, clean encoding in the Unicode standard, development of standards for the creation of an Amazigh keyboard and the creation of Tifinagh fonts. Since the creation of the Royal Institute of Amazigh Culture (IRCAM), the language has an institutional status allowing it to have an official spelling and clean in Unicode encoding. B.EL KESSAB and M.FAKIR have used in a work published in 2013 the ANN and the Markov model for the recognition of characters. Their approuche is tested on a database of characters Tifinagh script isolated Sizeable (900 images learning examples and 1275 test), the rate of reconnaisance found is 95.55% [2]. R.AYACHI and M.FAKIR have developed a system that uses the Fourier transform and invariant moments to extract attributes and dynamic programming and artificial neural networks for classification. The rate of reconnaissance found is in dynamic programming 92.76% and in ANN 93.27% [3]. A.Ouguengay he has a work of which he join the Tifinagh script engine Tesseract OCR Ocropus classification system. But the results are not as satisfactory [4]. K.MORO and B.KESSAB they made a contribution to the recognition of character Tifinaghe by rna and skeletonization and are obtained tau reconnisance of 92.52%. O.BENDCHAREF and N.IDRISSI they proposed the application of Riemannian geometry to the recognition of Tifinagh characters and they arrived to the rate of reconnaissance 93% [5].

1.3. Artificial Neural Network

The neural network is inspired by biological neural system. It is composed of several interconnected elements to solve a collection of varied problems. The brain is composed of billions of neurons and trillions of connections between them. The nerve impulse travels through the dendrites and axons, and then treated in the neurons through synapses.

This results in the field of artificial neural networks in several interconnected elements or belonging to one of the three marks neurons, input, output or hidden. Neurons belonging to layer n are considered an automatic threshold. In addition, to be activated, it must receive a signal above this threshold, the output of the neuron after taking into account the weight parameters, supplying all the elements belonging to the layer n +1. As biological neural system, neural networks have the ability to learn, which makes them useful.

The artificial neural networks are units of troubleshooting, capable of handling fuzzy information in parallel and come out with one or more results representing the postulated solution. The basic unit of a neural network is a non-linear combinational function called artificial neurons. An artificial neuron represents a computer simulation of a biological neuron human brain. Each artificial neuron is characterized by an information vector which is present at the input of the neuron and a non-linear mathematical operator capable of calculating an output on this vector. The following figure shows an artificial neuron [6]:

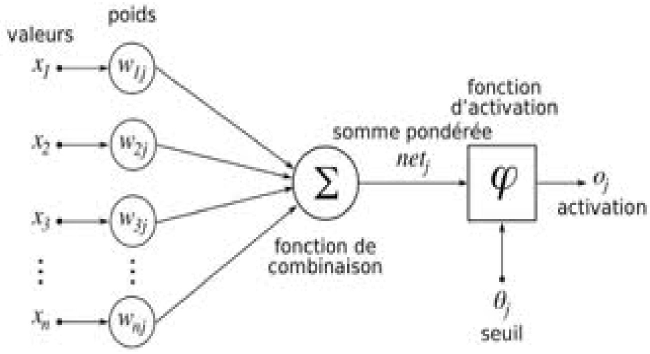


Figure 2. Artificial neuron

The synapses are Wij (weights) of the J neurons; they are real numbers between 0 and 1.

The function is a summation of combinations between active synapses associated with the same neuron. The activation function is a non-linear operator to return a true value or rounded in the range [0 1]. In our case we use the sigmoid function [7]:

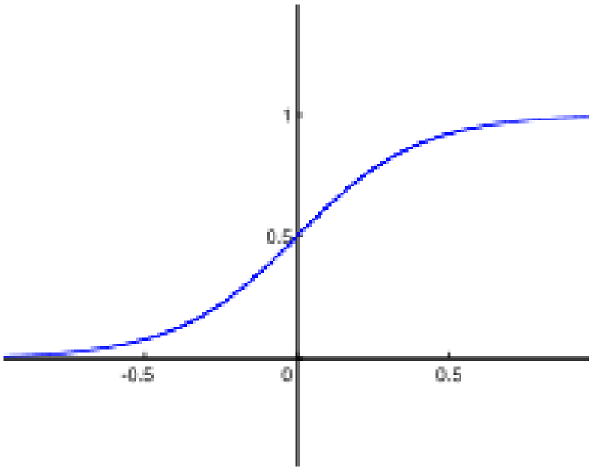


Figure 3. Sigmoid function

An ANN is composed by a collection of artificial neurons interconnected among them to form a neuronal system able to learn and to understand the mechanisms. Each ANN is characterized by its specific architecture; this architecture is denoted by the number of neurons of the input layer, the number of hidden layers, the number of neurons in each hidden layer and the neurons number in the output layer. A layer of neurons in a neural network is a group of artificial neurons, with the same level of importance, as is shown in the following figure [8]:

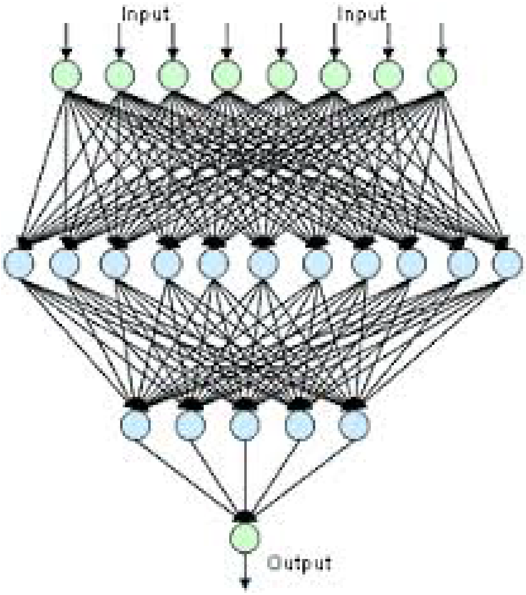


Figure 4. Artificial neural network

The operating principle of artificial neural networks is similar to the human brain; first, it must necessarily pass on the learning phase to record knowledge in the memory of the artificial neural network. The storage of knowledge is the principle of reputation and compensation to a collection of data that forms the basis of learning.

We have several algorithms that can teach an ANN as backpropagation. The backpropagation is a method of calculating the weight for network supervised learning is to minimize the squared error output. It involves correcting errors according to the importance of the elements involved in fact the realization of these errors: the synaptic weights that help to generate a significant error will be changed more significantly than the weights that led to a marginal error. In the neural network, weights are, firstly, initialized with random values. It then considers a set of data that will be used for learning.

1.4. Proposed Method

The main objective of this work is to analyze, design and implement a system that will allow character recognition of the Amazigh language using a probabilistic and statistical method based on the probabilistic calculation on text corpus of the Amazigh language through the construction of a database containing the frequency of use of each word of the Amazigh language. This database is based on a collection of corpus of the Amazigh language from several natures. The first phase of this work is the construction of the frequencies database. The second is the migration of data from the database in a method of character recognition based on neural networks. This method will allow us to reduce the number of processing operations associated with the recognition of characters; it will also allow us to maximize the rate of character recognition picture. In this work, we will also do a study to find all the morphological characteristics necessary and sufficient to represent the characters of the Amazigh language.

2. Methodology

We have a large collection of documents of the Amazigh language; these documents are officially provided by the Royal Institute of Amazigh Culture (IRCAM) to conduct the study in real standards and to maximize the degree of accuracy of the approach.

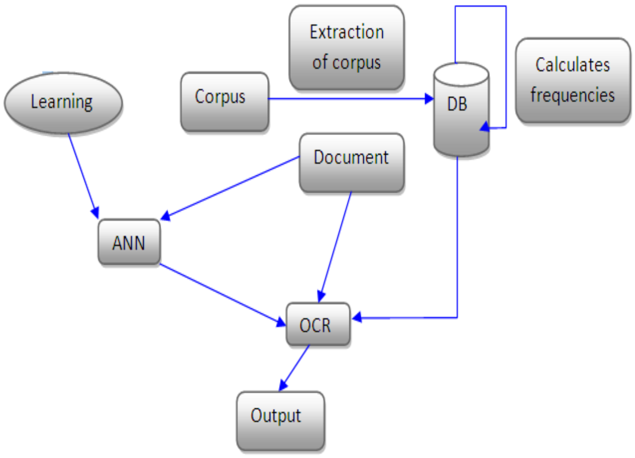


Figure 5. Principe of the method

In parallel with this collection of information, we build a MySQL database to study statistically the masses of documents to calculate the use frequency of each Amazigh. After the construction of the Tamazight frequencies database, we perform character recognition by extracting the eigenvectors characters and artificial neural networks.

The principle of this method is based on the following functional requirements:

1. The recognition and validation of the characters are done by the recognition and validation of a Word.
2. The recognition of all the characters of a Word is done by the partial recognition of the word.
3. The partial recognition of a Word is done by extraction of the eigenvectors and artificial neural networks.
4. Global recognition of the characters is done by the partial recognition of the most used words via frequencies database.

2.1. Corpus of Tamazight

The used corpus is a collection of artistic, scientific and other types of documents such as historical books, grouped in a specific context, in the aim to build a database of frequently used expressions in Tamazight. This database was recovered from the Royal Institute of Amazigh Culture (IRCAM) [11]. The majority of this corpus is not written with Amazigh characters, but they are transliterated in Arabic or Latin. For the sake of unification of the database construction and also for technical reasons, a phase of preparation and adaptation was established.

2.2. Building the Database

The frequencies database is a computer container for storing all the information related to the corpus Tamazight. This database contains mainly tables of lexical units and their frequency of use, and another table for the corpus and their frequencies. Table of tokens is used to word recognition, while the table corpus is useful to extend the system to the global recognition corpus. The database is composed mainly by the following tables:

- Table of lexical units:

```
   CREATE TABLE `frequence_base`.
`Unites_lexicales` (
      `ID` INTEGER UNSIGNED NOT NULL
AUTO_INCREMENT,
      `Code Unicode` VARCHAR(45) NOT NULL,
      `Code ASCII` VARCHAR(45) NOT NULL,
      `Code Arabe ` VARCHAR(45) NOT NULL,
      `Code latin` VARCHAR(45) NOT NULL,
      `Frequence` VARCHAR(45) NOT NULL,
      PRIMARY KEY (`ID`))
   ENGINE = InnoDB;
```

- SQL script: to create the frequencies table of Amazigh words.

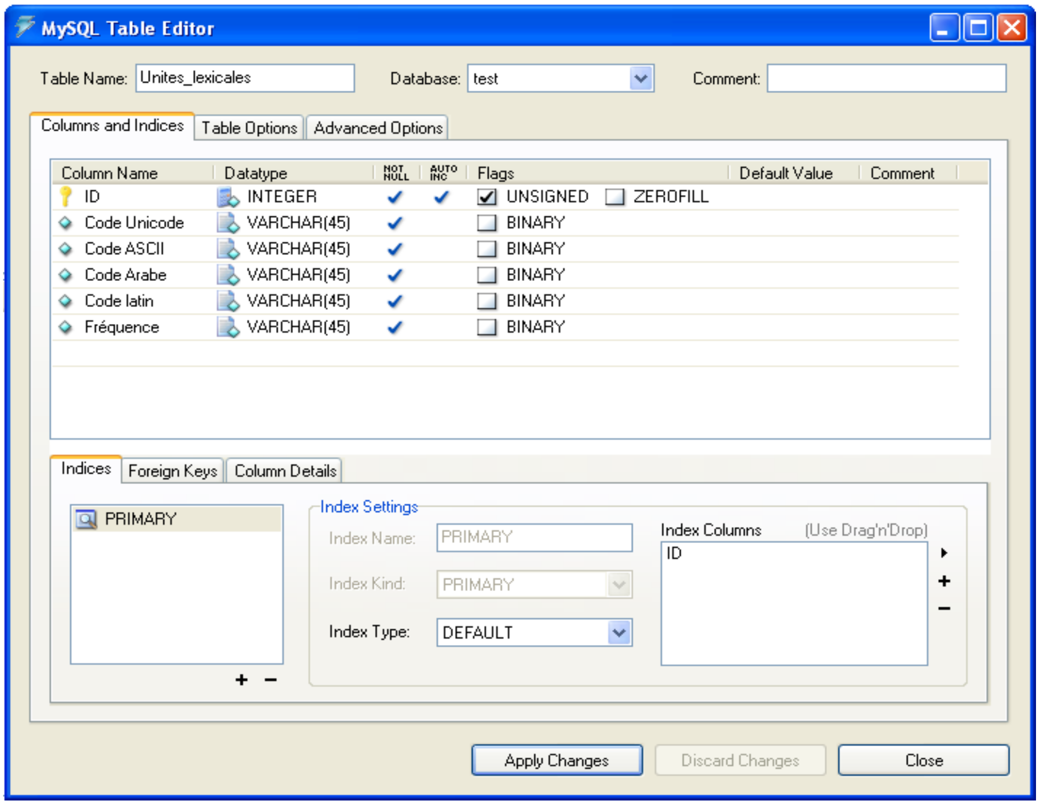


Table Name: Unites_lexicales Database: test Comment:

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
ID	INTEGER	✓	✓	☑ UNSIGNED ☐ ZEROFILL		
Code Unicode	VARCHAR(45)	✓		☐ BINARY		
Code ASCII	VARCHAR(45)	✓		☐ BINARY		
Code Arabe	VARCHAR(45)	✓		☐ BINARY		
Code latin	VARCHAR(45)	✓		☐ BINARY		
Fréquence	VARCHAR(45)	✓		☐ BINARY		

PRIMARY

Index Settings

Index Name: PRIMARY

Index Kind: PRIMARY

Index Type: DEFAULT

Index Columns (Use Drag'n'Drop)

ID

Figure 6. Table of lexical units

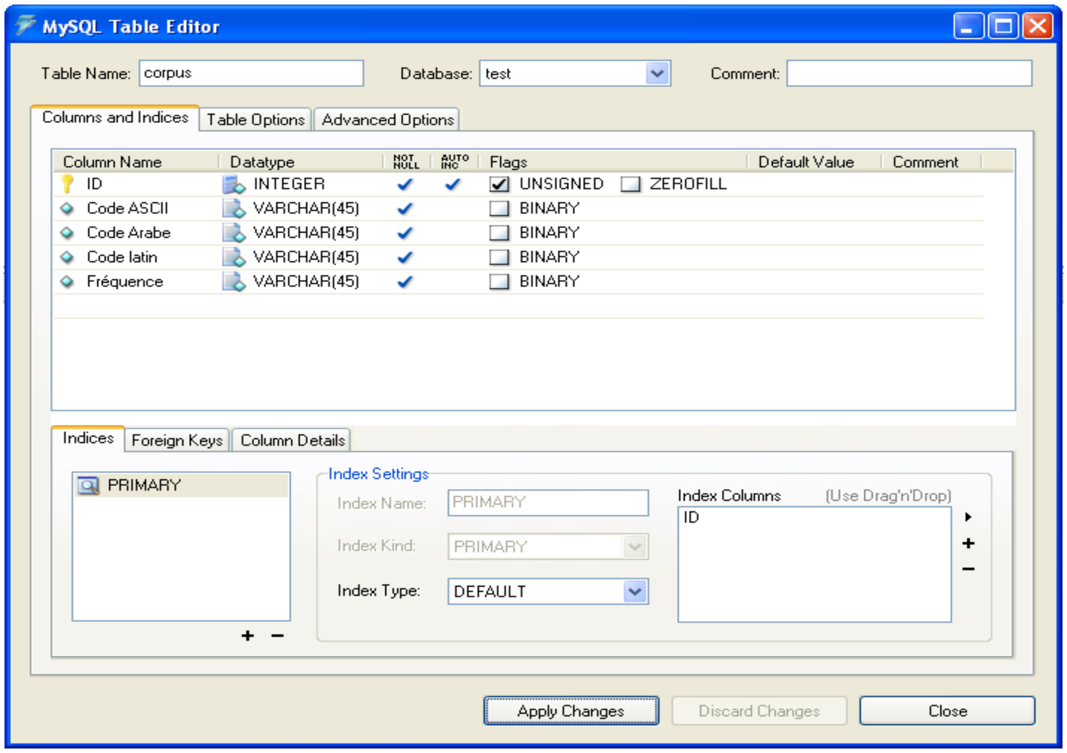


Table Name: corpus Database: test Comment:

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
ID	INTEGER	✓	✓	☑ UNSIGNED ☐ ZEROFILL		
Code ASCII	VARCHAR(45)	✓		☐ BINARY		
Code Arabe	VARCHAR(45)	✓		☐ BINARY		
Code latin	VARCHAR(45)	✓		☐ BINARY		
Fréquence	VARCHAR(45)	✓		☐ BINARY		

PRIMARY

Index Settings

Index Name: PRIMARY

Index Kind: PRIMARY

Index Type: DEFAULT

Index Columns (Use Drag'n'Drop): ID

Figure 7. Table of corpus

This table contains a unique identifier for each word, the ASCII code, IRCAM, Latin, Arabic word and word frequency.

- Table of corpus:

```
    CREATE TABLE ` frequence_base `.`corpus` (
      `ID` INTEGER UNSIGNED NOT NULL
AUTO_INCREMENT,
      `Code ASCII` VARCHAR(45) NOT NULL,
      `Code Arabe` VARCHAR(45) NOT NULL,
      `Code latin` VARCHAR(45) NOT NULL,
      `Fréquence` VARCHAR(45) NOT NULL,
      PRIMARY KEY (`ID`)
    )
    ENGINE = InnoDB;
```

- SQL script: to create the Amazigh corpus table.

The purpose of this database is to safeguard the statistical characteristics relating to corpus Tamazight and provide information engine that continuously supplies the recognition system. It should be noted that this database is constantly increasing thanks to a flexible system capable of analyzing new materials for the extraction of lexical Tamazight units and update the database.

2.3. Technical Specification of the Database

For automatic processing of Amazigh characters in programming languages such as C language, there is no possibility to directly handling the word of Tamazight, because we do not have C compiler accepts the implementation of these characters. Therefore, we will proceed to make a transliteration of Tamazight words in other words written by Latin characters.

The Royal Institute of Amazigh Culture has proposed standards for standardization of Unicode character codes IRCAM. As a result, we will use these codes to safeguard the lexical units and corpus of Tamazight in the database in order to facilitate migration in the future, and in parallel, we will also save them in formats transliterated for reasons of simplification processing all other documents and also to enjoy the power of GCC and Flex compiler for the treatment of Latin characters. This database runs on the management system MySQL database.

2.4. Calculating Frequencies Algorithm

There is an open and ready database which save records generated by the algorithm for calculating frequencies, and has a collection of documents Tamazight PDF, doc and txt.

The algorithm makes lexical analysis of each document presented to the input, then it is the recognition of a lexical unit (word or corpus) and finally launches a SQL query to the database to check whether this frequency token already exists in the database:

If it already exists in the database then the frequency of use of the lexical unit (the number of occurrences and the lexical unit divided by the total number of tokens processed on all documents) one updated.

If the token does not exist in the database (in the case corresponding to the first occurrence of the token) then add this into the base unit and its frequency is initialized.

It is clear that the calculation algorithm is a frequency -linear algorithm based on the number of lexical items presented to the input. Since, we have for each word or phrase for each algorithm will launch a racket on the database to derive the number of occurrences which reflects clear the power of this mechanism updated in real-time data based.

2.5. Program to Build the Frequency Base

Flex is an ideal and very powerful, tool for automatically generating lexical analyzers based on the description of lexical initiated by the definition of regular expressions. The program of construction of the base frequency is a flex program composed of three sections:

1. The definition section of macrous programming and description of lexical units and import libraries.
2. The rules section UNITS lexical production and communication with the database.
3. The program committee section results in the database and destroy the connection with the base. Among the main lexical units that must be delivered , we have:
4. MOT: as a result of concatenation of characters ([a -zA- Z]).
5. CORPUS: as a result of MOT separated by the space character (([PASSWORD] " [PASSWORD])).

Flex compiling this program will generate a lexical analyzer coded C programming language capable of supplying the database.

2.6. IRCAM Character Recognition by Artificial Neural Networks

In this section, we will expose a method for recognizing characters IRCAM with artificial neural networks based on the extraction of the eigenvectors of the morphological characteristics of each character IRCAM.

2.7. General Description of the Process

The objective of this method is to create an artificial neural network, with a suitable architecture, which learns to recognize characters using eigenvectors of these ones. At first, we must make a selective search on the really important and technically easy to extract and less costly in terms of the complexity characteristics to form the basis of learning artificial neural network. Then, another search to select the artificial neurons network architecture most suitable (which minimizes convergence time and maximizes the recognition rate). And finally, to accelerate the recognition process, we must seek a relationship between the weight of each feature and the activation of the neuron associated with this feature input.

2.8. Characteristics of IRCAM Characters

Each character is IRCAM morphologically identified by several characteristics (characteristics over 80 [9]) whose potential varies between them. The following figure shows the features that we used [10]:

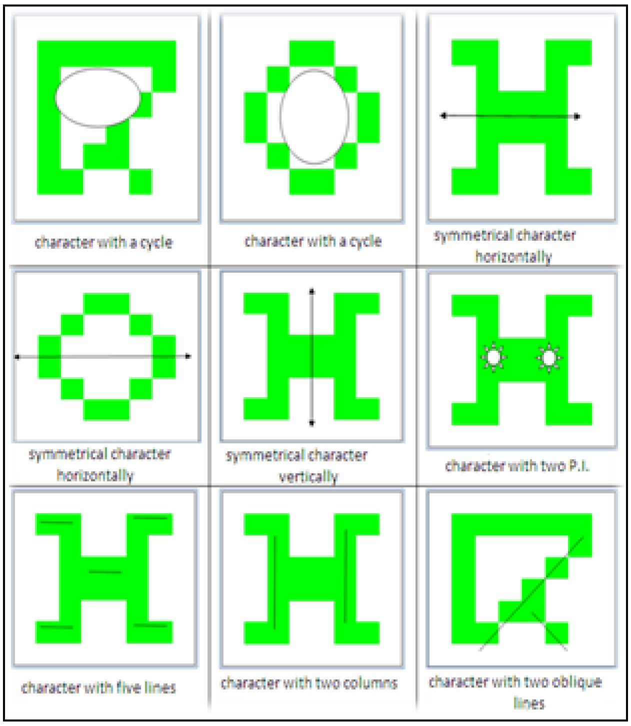
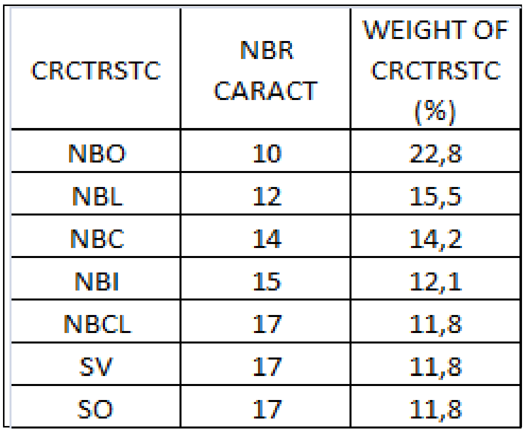

character with a cycle
character with a cycle
symmetrical character horizontally
symmetrical character horizontally
symmetrical character vertically
character with two P.I.
character with five lines
character with two columns
character with two oblique lines

Figure 8. Characteristics of IRCAM characters

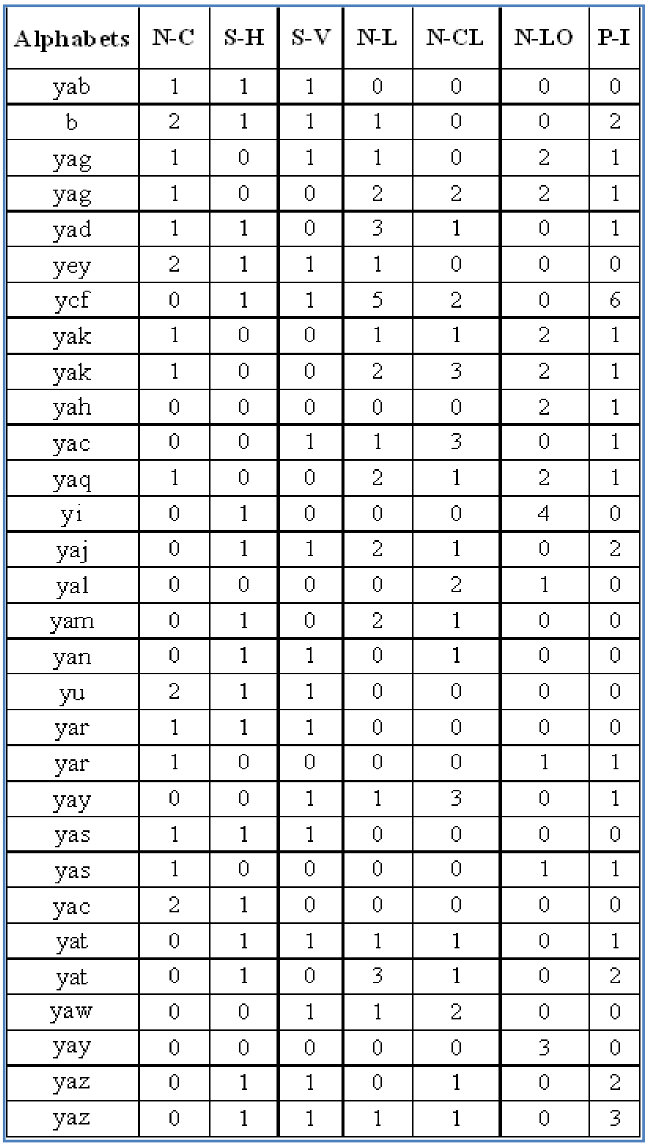
We made the extraction of a specific characteristic of each character in vector 7 cells:



CRCTRSTC	NBR CARACT	WEIGHT OF CRCTRSTC (%)
NBO	10	22,8
NBL	12	15,5
NBC	14	14,2
NBI	15	12,1
NBCL	17	11,8
SV	17	11,8
SO	17	11,8

Figure 9. Characteristics of IRCAM characters

We can easily prove that this vector broadly covers the wholes IRCAM characters:



Alphabets	N-C	S-H	S-V	N-L	N-CL	N-LO	P-I
yab	1	1	1	0	0	0	0
b	2	1	1	1	0	0	2
yag	1	0	1	1	0	2	1
yag	1	0	0	2	2	2	1
yad	1	1	0	3	1	0	1
yey	2	1	1	1	0	0	0
ycf	0	1	1	5	2	0	6
yak	1	0	0	1	1	2	1
yak	1	0	0	2	3	2	1
yah	0	0	0	0	0	2	1
yac	0	0	1	1	3	0	1
yaq	1	0	0	2	1	2	1
yi	0	1	0	0	0	4	0
yaj	0	1	1	2	1	0	2
yal	0	0	0	0	2	1	0
yam	0	1	0	2	1	0	0
yan	0	1	1	0	1	0	0
yu	2	1	1	0	0	0	0
yar	1	1	1	0	0	0	0
yar	1	0	0	0	0	1	1
yay	0	0	1	1	3	0	1
yas	1	1	1	0	0	0	0
yas	1	0	0	0	0	1	1
yac	2	1	0	0	0	0	0
yat	0	1	1	1	1	0	1
yat	0	1	0	3	1	0	2
yaw	0	0	1	1	2	0	0
yay	0	0	0	0	0	3	0
yaz	0	1	1	0	1	0	2
yaz	0	1	1	1	1	0	3

Figure 10. Characteristics of all IRCAM characters

We make a selection of those seven characteristics among others in order to minimize the treatment time of documents (recognition time).

2.9. ANN Architecture

A comparative study is carried out to select the most efficient architecture in terms of learning time and recognition rate of artificial neural network, this study is to instantiate multiple artificial neural networks with different architectures. The architecture is defined by a multiple of hidden layers and a number of artificial neurons in each layer.

The instantiation of these neural networks is generated with a description meta-module integrated into a system of automatic generation of artificial neural networks that we have developed for this objective.

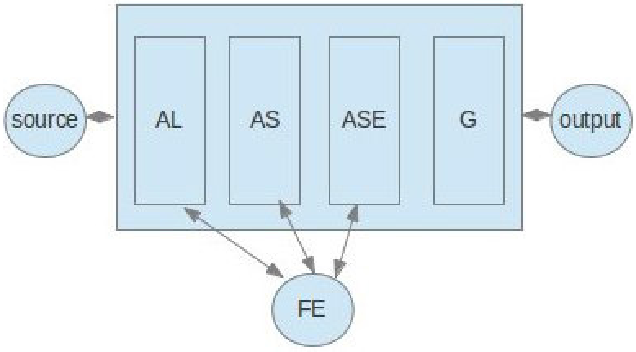


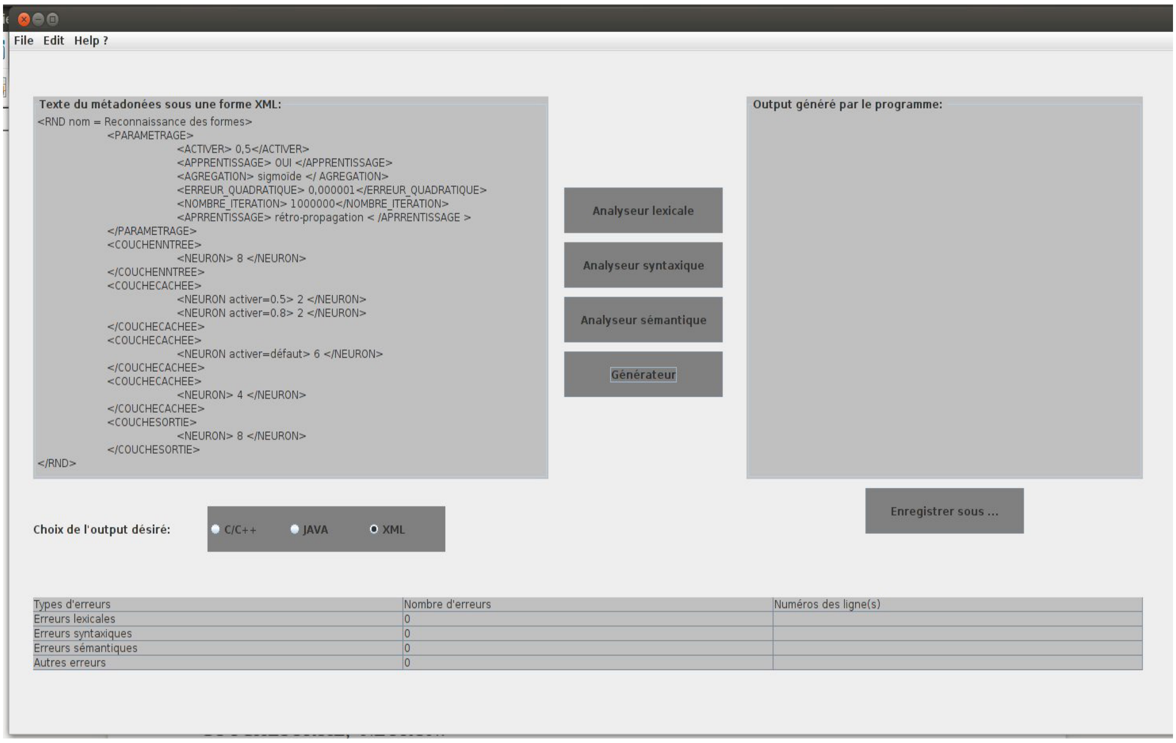
Figure 11. General Architecture of the automatic generator artificial neural network

The generator is composed mainly by four programs:

1. Lexical analyzer: who makes the verification of the lexicon and the recognition of all lexical units and fills the error file if any errors ever met. If the lexical analyzer completes his execution without encountering any error then it passes to parser the program, it should be noted that this latter is generated by the Flex tool that provides a C program that must be compiled to give an executable.
2. The parser can recognize syntax errors then recorded them in the error log and evaluate the expressions using the units generated by the lexical analyzer program.
3. The semantic analyzer allows for the semantic validation of sources according to all specifications of artificial neural networks. If we encounter errors, we have to save them in the errors file and stop the build operation.

Syntactic, lexical and semantic analyzers represent the control phase of the generator while the generator program do the automatic generation of a program written in an object oriented language. It should be noted that this application can generate an ANN program directly in C++ and java or in an XML form that must be exploited to expand the set of target languages.

After a series of tests and research, we selected the following experimental architecture as good architecture: 7 neurons in the input layer, one hidden layer with 4 neurons and an output layer with 7 neurons. For the input layer, we took 7 because it will use an eigenvector of 7 cells, and since learning is supervised so we took 7 neurons in the output layer.



Texte du métadonées sous une forme XML:

```
<RND nom = Reconnaissance des formes>
        <PARAMETRAGE>
                <ACTIVER> 0,5</ACTIVER>
                <APPRENTISSAGE> OUI </APPRENTISSAGE>
                <AGREGATION> sigmoïde </ AGREGATION>
                <ERREUR_QUADRATIQUE> 0,000001</ERREUR_QUADRATIQUE>
                <NOMBRE_ITERATION> 1000000</NOMBRE_ITERATION>
                <APRRENTISSAGE> rétro-propagation < /APRRENTISSAGE >
        </PARAMETRAGE>
        <COUCHENNTREE>
                <NEURON> 8 </NEURON>
        </COUCHENNTREE>
        <COUCHECACHEE>
                <NEURON activer=0.5> 2 </NEURON>
                <NEURON activer=0.8> 2 </NEURON>
        </COUCHECACHEE>
        <COUCHECACHEE>
                <NEURON activer=défaut> 6 </NEURON>
        </COUCHECACHEE>
        <COUCHECACHEE>
                <NEURON> 4 </NEURON>
        </COUCHECACHEE>
        <COUCHESORTIE>
                <NEURON> 8 </NEURON>
        </COUCHESORTIE>
</RND>
```

Choix de l'output désiré: ○ C/C++ ○ JAVA ● XML

Types d'erreurs	Nombre d'erreurs	Numéros des ligne(s)
Erreurs lexicales	0	
Erreurs syntaxiques	0	
Erreurs sémantiques	0	
Autres erreurs	0	

Figure 12. Automatic generator artificial neural network

2.10. Weight of Characteristics and Activation of Artificial Neurons

We made the selection of morphological characteristics for use in the learning and recognition, and we selected the architecture of the ANN most suitable for the problem of IRCAM characters recognition. Finally, we noticed that these features do not have the same potential. Then we conducted another study to determine the potential of each character we have shown this problem with the following graph:

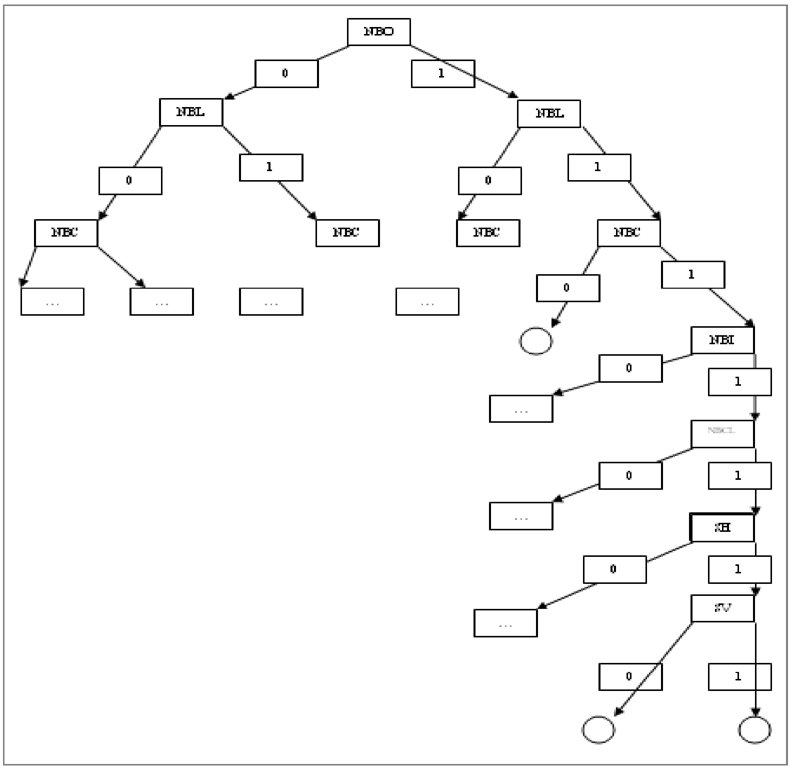


Figure 13. Tree of recognition

We have exploited the potentials of characteristics in order to change the learning algorithm used (neurons whose potential is below a given threshold is deactivated).

2.11. Learning Algorithm

Step 1: Initialize the connection weights (weights are taken randomly).

Step 2: Propagation entries

Entered the E_i are presented to the input layer:

$$X_i = E_i \; ; \qquad (1)$$

The spread to the hidden layer is made using the following formula:

$$Y_i = f\left(\sum_{i=1}^{7} X_i * W_{ji} + X_0\right); \qquad (2)$$

Then from the hidden layer to the output layer, we adopt:

$$Z_k = f\left(\sum_{i=1}^{3} Y_i * W_{kj} + Y_0\right); \qquad (3)$$

X_0 and Y_0 are scalar.

$f(x)$ is the activation function(sigmoid):

$$f(x) = \frac{1}{1+e^{-x}} \; ; \qquad (4)$$

Step 3: Back propagation of error at the output layer, the error between the desired output S_k and Z_k output is calculated by:

$$E_k = Z_k(Z_k - 1)(S_k - Z_k) \; ; \qquad (5)$$

The error calculation is propagated on the hidden layer using the following formula:

$$Y_j = f_j\left(1 - Y_j\right)\sum_{k=1}^{7} W_{kj} \; E_k \; ; \qquad (6)$$

Step 4: Fixed connection weights the connection weights between the input layer and hidden layer is corrected by:

$$DW_{ji} = n \, X_i F_j ; \qquad (7)$$

And

$$DY_0 = n \, F_j \; ; \qquad (8)$$

then we were changing the connections between the input layer and the output layer by:

$$DW_{kj} = n \, Y_j E_k \; ; \qquad (9)$$

N is a parameter to be determined empirically.

Step 5 loops: Loop to step 2 until a stop to define criterion (error threshold, the number of iterations).

2.12. Partial Recognition of a Word

The edict of this work is not to make the classic recognition of each character that makes up the word, but to do the recognition that a portion (two or three characters) of the word. The objective is to reduce the number of image processing operations on the documents. Then the database is used to derive frequency and recognize the word.

a) Overall character recognition

After the partial recognition of a word with the use of the database frequencies and artificial neural networks, we deduce the whole other characters in the word.

b) General architecture of the recognition system

The following figure shows the relationship between the sub-programs who constitute the recognition system.

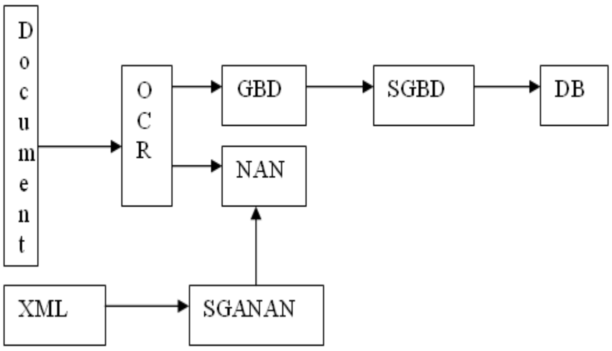


Figure 14. General architecture of the recognition system

1. BD: indicates the database frequency words and corpus.
2. GBD: this is the program that will make the extraction of words and corpus then calculated the frequencies and updated the database.
3. CORPUS: is a collection of articles Tamazight.
4. XML: specifies the file that contains the description of the structure of ANN used for character recognition as a meta-module.
5. SGARNA: the automatic generation system of the structure of a neural network from a description meta-module.
6. RNA: the ANN generated by the automatic generation of the structure of artificial neural network. It is this network that will make the partial recognition of characters in a word:

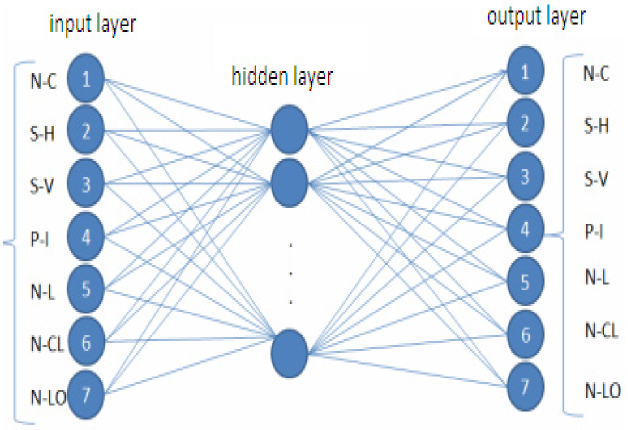


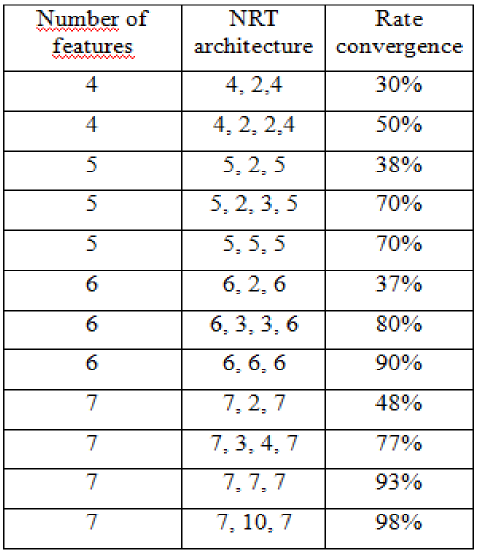
Figure 15. OCR artificial neural network

OCR: this is the main program that will start the process of recognition of characters per word recognition. Documents: the documents containing the text to be recognized.

3. Results

In this section, we will present the results we have found the test data used. As a first step, we will discuss the results of recognition using only the eigenvector characters and ANN as a classification approach, and in the following, we will show the results of using the hybrid probabilistic calculation and classification by artificial neural networks.

- With artificial neural networks:

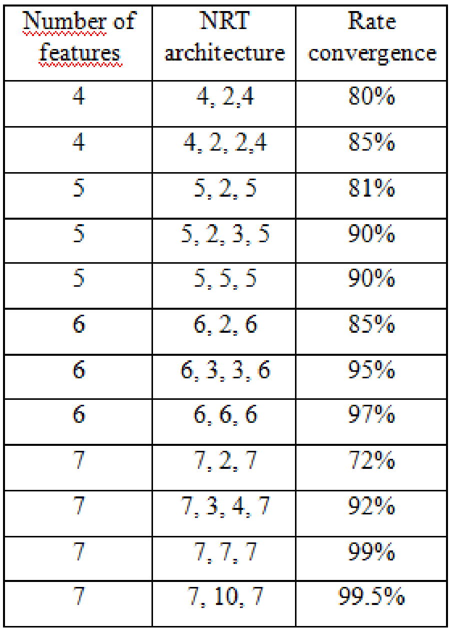


Number of features	NRT architecture	Rate convergence
4	4, 2,4	30%
4	4, 2, 2,4	50%
5	5, 2, 5	38%
5	5, 2, 3, 5	70%
5	5, 5, 5	70%
6	6, 2, 6	37%
6	6, 3, 3, 6	80%
6	6, 6, 6	90%
7	7, 2, 7	48%
7	7, 3, 4, 7	77%
7	7, 7, 7	93%
7	7, 10, 7	98%

Figure 16. Results with artificial neural networks only

In the case of using the ANN reduced, we have a 98% rate of recognition, with the architecture (7, 10, 7).

- With more artificial neural networks using database frequencies:



Number of features	NRT architecture	Rate convergence
4	4, 2,4	80%
4	4, 2, 2,4	85%
5	5, 2, 5	81%
5	5, 2, 3, 5	90%
5	5, 5, 5	90%
6	6, 2, 6	85%
6	6, 3, 3, 6	95%
6	6, 6, 6	97%
7	7, 2, 7	72%
7	7, 3, 4, 7	92%
7	7, 7, 7	99%
7	7, 10, 7	99.5%

Figure 17. Results with artificial neural networks and database frequencies

So, after these results, we can see that the network architecture artificial neurons (7, 10, 7) is the perform one to the problem of Tifinagh OCR. The recognition rate found is around 99.5%. The important difference between these two methods is not visible in the rate of recognition, but it is clearly visible in the time to rethink programs.

4. Conclusions and Perspectives

Such an approach of complete classification can't be made on the IRCAM characters, while the combination algorithms and the optimization algorithms of image processing and feature extraction remains a very open research problem in order to strengthen the classification process.

In this work we presented an approach based on the probabilistic and the artificial neural networks to eliminate the problems of unclassified characters increase the recognition rate (worst classification of characters) and most importantly reduce the recognition time.

Still using the same principle we can do the character recognition by a deduction of recognized corpus. We can also add a validation test corpus with a summation over the probabilities of words in the corpus.

REFERENCES

[1] http://www.ircam.ma/fr/index.php?soc=telec.

[2] B.EL KESSAB and M.FAKIR, using neural networks and Markov model for reconnisance of characaters Tifinagh manuscripts, SITACAM’09, 2010.

[3] R.EL AYACHI M.FAKIR, Fourier transform and time invariant applied to the recognition of characters Tifinaghe , SITACAM’09, 2010.

[4] A.Ouguengay, Integration of the Amazigh in OCR Open Source as a model Ocropus, SITACAM’09, 2010.

[5] O.BENDCHAREF and N.IDRISSI, Riemannian geometry to the recognition of Tifinagh characters. SITACAM’09, 2010.

[6] http://upload.wikimedia.org/wikipedia/commons/thumb/5/58/ArtificialNeuronModel_francais.png/420px-ArtificialNeuronModel_francais.png

[7] http://upload.wikimedia.org/wikipedia/commons/thumb/9/9d/Sigmoide.PNG/220px-Sigmoide.PNG

[8] https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcS_dosVDqdK7WZI4aNFSaDNZxjQJhVirSOFdMQIzFK1Rk1kxNsg.

[9] Desai (2010), Gujarati handwritten numeral Optical Character Reorganization through neural network pattern recognition 43-2010.

[10] H.TARIK, J. EL MILOUD, Tifinagh characters recognition by artificial neural network, SITACAM’13, 2012.

[11] R. EL Yachi, K. Moro, Mr. Fakir, B. Bouikhalene (2010), Evaluation Methods SKeletonization for the recognition of printed Tifinaghe Characters, Sitacam'09.

[12] http://www.ircam.ma/fr/index.php?soc=ircam&rd=2.

[13] Mr. Banoumi, A. Lazrek and K.Sami (2002), an Arabic transliteration / novel for e-paper, Proceeding of the 5th International Symposium on Electronic Document, Federal Conference on the document, Hammamet, Tunisia.

[14] C. Rey & J.-L. Dugelay. Blind Detection of Malicious Alterations On Still Images Using Robust Watermarks. IEE Secure Images and Image Authentication colloquium, London, UK, Apr. 2000.

[15] M. Kutter, S. Voloshynocskiy and A. Herrigel. The Watermark Copy Attack. In Proceedings of SPIE Security and Watermarking of Multimedia Content II , vol. 3971, San Jose, USA, Jan. 2000.

[16] J. Fridrich, M. Goljan & N. Memon. Further Attacks on Yeung-Minzer Fragile Watermarking Scheme. SPIE International Conf. on Security and Watermarking of Multimedia Contents II , vol. 3971, No13, San Jose, USA, Jan. 2000.

[17] Julien Chable. Emmanuel Robles Programmation Android, de la conception au déploiement avec le SDK Google Android, New York, France, 2009.

[18] UML Infrastructure Final Adopted Speci_cations, version 2.0, Sebtembre 2003. http://www.omg.org/cgi-bin/doc ?ptc/03-09-15.pdf.

[19] UML Infrastructure Final Adopted Speci_cations, version 2.0, Sebtembre 2003. http://www.omg.org/cgi-bin/doc ?ptc/03-09-15.pdf.

[20] S. Bhattacharjee and M. Kutter. Compression Tolerant Image Authentication. IEEE International Conf. on Image Processing (ICIP'98), Chicago, USA, Oct. 1998.

[21] J.-L. Dugelay & S. Roche. Introduction au tatouage d'images. Annales des Télécommunications, 54, no 9-10, pp. 427-437, 1999.

[22] J.-L. Dugelay. Procédé de dissimulation d'informations dans une image numérique. Brevet INPI FR 98-04083 (EURECOM 09-FR), March 1998.

[23] J.-L. Dugelay & S. Roche. Process for marking a multimedia document, such an image, by generating a mark. Pending patent EP 99480075.3 (EURECOM 11/12 EP), July 1999.

[24] J. Fridrich. Robust Bit Extraction From Images. ICMCS'99, Florence, Italy, june 1999.

[25] J. Fridrich. Methods for detecting changes in digital images. Proceedings IEEE Int. Conf. on Image Processing (ICIP'98), Chicago, USA, Oct. 1998.

[26] Kodak. Understanding and Intergrating KODAK Picture Authentication Cameras. http://www.kodak.com/US/en/digital/software/imageAuthentication/

[27] D. Kundur and D. Hatzinakos. Towards a Telltale Watermarking Technique for Tamper-Proofing. IEEE International Conf. on Image Processing (ICIP'98) , Chicago, USA, Oct. 1998.

[28] C.-Y. Lin and S.-F. Chang. A Watermark-Based Robust Image Authentication Using Wavelets. ADVENT Project Report, Columbia University, Apr. 1998.