Recognition of Amazigh language transcribed into Latin based on polygonal approximation

K. EL GAJOUI, F. ATAA ALLAH, M. OUMSIS

Abstract**—**Optical Character Recognition systems aim to achieve a complete conversion of document image to fully searchable text file. These systems are composed of a set of modules. Different approaches have been developed for each module and each approach includes several techniques that have been suggested by different researchers. In this paper, we propose an OCR system for dealing with the Amazigh language, transcribed into Latin letters with diacritical marks. To this aim, we created an OCR corpus associated to this language, we used nonlinear binarization method in the preprocessing phase, and we adopted a structural approach based on polygonal approximation features for the classification phase.

Keywords**—**Amazigh, Diacritical marks, OCR, Structural approach.

I. Introduction

THE optical character recognition, or OCR, is an information technology (IT) process that enables to recognize letters in a text image file, and converts it into a text file [1], [2]. The main advantage of this technique is being able to take full benefit of the electronic version of a textual document: edit, index, search in a text, as well as select words or sentences from the same text. Therefore, the OCR processing is very useful, especially, for processing documents available only in printed version.

The Amazigh language is spoken by a large number of populations in North Africa. Recently, it was made an official language of Morocco, along with Arabic, and has undergone a tangible revolution in research. In order to preserve the Moroccan Amazigh language, save its literary heritage, and digitalize its old documents, one of the axes of research that have been developed for this language is OCR. Since its old documents were transcribed in Latin and Arabic letters, we have focused, in this work, on Latin characters, especially, that the most existing systems for Amazigh focused on Tifinagh writing [3], [4].

In the remaining of this paper, we introduce the OCR system architecture, in Section 2. In Section 3, we present a state of the art of OCR. In Section 4, we introduce the steps for creating the Amazigh OCR corpus. In Section 5, we describe our proposed system. Then, we show, in Section 6, the evaluation of the system tested on a set of documents extracted from different books. Finally, in Section 7, we draw conclusions and suggest further related research.

II. Optical character recognition systems

The OCR systems have several architectures. They vary from one system to another as needed. We can generalize all the proposed architectures by the representation illustrated in Fig 1, since the OCR systems are usually composed of the following phases [5]:

1) Acquisition allows the conversion of a printed document to a numerical image format via capture equipments.
2) Preprocessing phase: prepares the sensor data to the next phase. It includes a set of treatments applied to the image in order to improve its quality.
3) Segmentation phase delimits document elements (line, word, character, ...). Good text segmentation plays an important role in increasing the OCR systems' performance.
4) Feature extraction phase describes various features characterizing delimited elements of a document. It is one of the most important steps in developing an OCR system.
5) Classification phase recognizes and identifies each element. It is performed based on the extracted features.
6) Post-processing phase increases the recognition rate by correcting errors. It is an optional phase and may be automatic or manual.

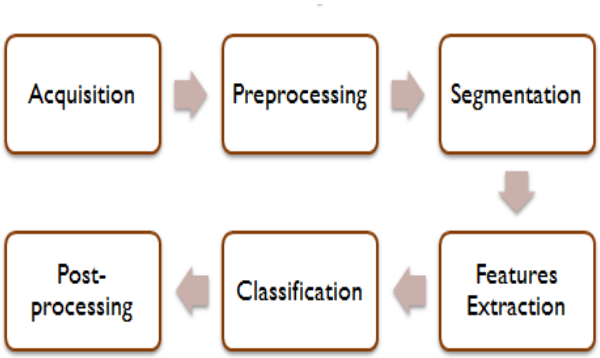


Fig. 1 General steps of OCR Systems

III. State of the Art

The OCR systems comprise a set of modules. During the last decades, several approaches and technologies have been developed for each module [6].

A. Preprocessing

In order to increase the chances of a good recognition, some pretreatments are necessary. Among these pretreatments, we quote:

- Binarization

Binarization is an operation that produces, by using the thresholding approaches, two classes of pixels. These classes are represented by black and white pixels.

In this context, two techniques have been developed:

- Global approach or single threshold, known by Otsu method [7]. It is calculated from a global measurement over the entire image.
- Local approach or local binarization, for which the classification of a pixel depends on the pixel itself and on its local information.

- Skew detection and correction

The detection of the skew angle of inclination is a common operation in analyzing documents. It is often due to incorrect positioning of the document on the scanner, resulting from an image inclination.

Several methods for evaluating the inclination angle have been proposed. However, the Hough transform is the most used [2], [7].

- Noise removal

Several techniques are used for noise elimination. Among these techniques, we can mention the application of different types of filters and the extraction of small connected components [5], [7], [8].

- Normalization

Normalization allows setting the characters of an image into standard sizes [8], [9].

- Skeletonization

The purpose of this technique is to simplify the shape of the characters in an image into a set of lines easy to process. It consists in reducing the character into its outline [10]. The skeleton must preserve the shape, connectivity, topology, and the ends of the route. Moreover, it should not introduce parasitic elements.

B. Segmentation

The aim of this phase is to separate text blocks from graphic blocks and extract from each text block lines, then words, characters and pseudo-characters, as needed, from the extracted lines.

To this end, two approaches have been developed:

- Global approach

It is based on the whole word as an indivisible entity characterized by a wealth of information allowing it to easily absorb variations in writing.

However, the general aspect of this approach limits it to distinct and reduced vocabularies [1].

- *Analytical approach*

It is based on a division (segmentation) of the word. This approach is used in the case of large vocabularies [1].

Several techniques can be applied:

- Connected component: This technique is based on the extraction of connected components [7].
- Vertical and horizontal histogram [3]: The information contained in histogram is represented in two axes. The horizontal axis expresses the change in levels of 0 (black) to 255 (white), and the vertical axis gives the number of corresponding pixels at each level.
- Frame or fixed-size windows [8]: The principle of this technique is to scan the image word by word with a window (frame). Each frame is transformed into a sequence of feature vectors calculated from a sliding window of NxN pixels that moves with M pixels from right to left or from left to right.

C. Features extraction

The purpose of feature extraction step is to extract relevant properties of the segmented elements. This phase represents one of the most difficult problems in pattern recognition [1].

To deal with this problem, the straightforward way is to describe the real matrix of characters. Nevertheless, there is an easiest approach, based on the extraction of features that characterize relevant elements and omits irrelevant attributes.

These features can be classified into the following types:

➢ Points distribution

- Zoning: The densities of black points inside regions are calculated and used as features.
- Distance: The characteristics are represented by the number of times that the vectors cross the character shape along certain directions.
- Moment: The moments of black points from a selected center are used as features. The most used methods for calculating the moment are: Invariant Moment and the Modified Invariant Moment [3].

➢ Transformation

This technique helps to reduce the dimensionality of the features' vector [1]. The extracted features are invariant to global deformations such as translation and rotation. Some of the transformation functions used are Fourier, Walsh, and Hough.

➢ Structural Analysis

The characteristics describing the geometry and structural topology of characters are extracted.

With these characteristics, we seek to describe the physical composition of the character. The most commonly used features are bays, endpoints, intersections between the lines and loops, in addition to the polygonal approximation segments [1], [9], [10].

D. Classification

Classification is the process of identifying each character by assigning it to the correct class. It helps to decide on the identity of a character by learning its form. Two kinds of approaches have been developed in this context:

➢ Statistical approach

Statistical approaches are based on the statistical study of measurements of the shapes to be recognized [1]. The study of their distribution in a metric space and statistical characterization of classes allows taking a recognition decision [8].

In the following, we quote five statistical methods among the most commonly used ones:

- Bayesian approach: It consists in selecting from a set of characters one for which the series of the extracted primitives have the highest posterior probability relative to the characters previously learned.
- K Nearest Neighbor method (KNN): The KNN algorithm compares the unknown form with forms stored in a reference class named prototype and assigns the class to its closest.
- Neural Networks (NN): They are composed of simple connected elements or neurons. These elements are strongly inspired by the biological nervous system [3], [7], [11].
- Hidden Markov Model (HMM): It is a probabilistic method whose model consists of a set of states, transitions probabilities between these states and observations made by the system on an image. These observations are represented by random variables, whose distribution depends on the state [10].
- Support Vector Machines (SVM): They are a group of supervised learning methods for classification. The classification usually uses training and testing data sets. The standard SVM classifier produces a model based on the training data, then takes the set of test data and predicts to classify them in one of the only two distinct classes. [12].

➢ Structural approach

Structural methods are based on the physical structure of characters. They seek to find simple or primitive elements of topological type (a loop, bow, ...) and describe their relations [1], [8].

Among the structural methods, we can mention:

- Test methods: They consist in applying tests on each character concerning the presence or absence of single elements or primitives to determine its class.
- Line comparison: characters are represented by line of primitives. Comparison of characters, treated with the reference model, consists in measuring the similarity between two lines and deciding on them. The measure of similarity can be done by calculating the distance or by examining the inclusion of all or a part of a chain in the others.

IV. CORPUS CONSTRUCTION

The Amazigh language, or Tamazight, is present today in a dozen of countries across the Maghreb-Sahel-Sahara: Morocco, Algeria, Tunisia, Libya, Egypt, Niger, Mali, Burkina Faso and Mauritania. Algeria and Morocco are by far the two countries with the largest Amazigh population.

Since antiquity, Amazigh people have their own writing system, called Tifinagh [13]. However, when it comes to write consistent documents, they used languages and / or scripts of dominant peoples in contact, such Punic, Latin or Arabic.
In Morocco, three writing systems are used to transcribe Amazigh language [14]:

- Tifinagh is the authentic alphabet, attested in Libyan inscriptions since antiquity, and the official script in Morocco for writing Amazigh language, since 2003.
- Arabic alphabet used since the Arab arrival on the 6th century.
- Latin characters used since the end of the 19th century by colonial scholars, and later by national researchers.

In this work, we focus on Latin transcription.

To create a corpus associated with the Amazigh language transcribed in Latin, we were based on a set of documents written in this language and based on rich and diverse

transcription charsets.

After exploring a set of Amazigh documents transcribed in Latin, such as "CHOIX DE VERSION BERBERES PARLER DU SUD-OUEST MAROCAINE" by Arsène Roux [13] "MOTS ET CHOSES BERBERES" by Emile Laoust [15] and "THE ARGAN TREE AND ITS TASHELHIYT BERBER LEXICON" by Harry Stroomer [16], we notice the following remarks:

- The studied language is a diacritical language.
- The characters used in the transcription are represented in Latin, Extended-A Latin and Extended Additional Latin encoding blocks.
- The global number of the characters used is 115 elements containing Latin characters in upper and lower case, and non letter characters.
- These characters are composed of Latin alphabet and diacritics that represent a set of marks accompanying a letter or grapheme.
- The diacritics used are placed above (superscript diacritic), below (subscribed diacritic) or after (adscript diacritic).

The figure and table below show respectively an example of text written in Amazigh language transcribed into Latin and an example of characters used in the transcription.

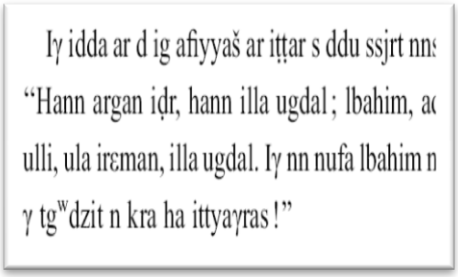

Iɣ idda ar d ig afiyyaš ar iṭṭar s ddu ssjrt nns
"Hann argan iḍr, hann illa ugdal; lbahim, ac
ulli, ula irɛman, illa ugdal. Iɣ nn nufa lbahim n
ɣ tgʷdzit n kra ha ittyaɣras!"

Fig. 2 An example of text excerpted from the book " THE ARGAN TREE AND ITS TASHELHIYT BERBER LEXICON"

Table I An example of characters used in Amazigh transcription into Latin

Ā	ā	Ă	ă	Aᶜ	aᶜ	Bʷ	bʷ	Bᶜ	bᶜ
Ḅ	ḅ	Ḍ	ḍ	Dᶜ	dᶜ	Ě	ě	Fʷ	fʷ
Ġ	ġ	Gʷ	gʷ	Ḡ	ḡ	Ḥ	ḥ	Ḫ	ḫ
Kʷ	kʷ	Lᶜ	lᶜ	Ḷ	ḷ	Mʷ	mʷ	Ǒ	ǒ
Ṛ	ṛ	Ṡ	ṡ	Tᶜ	tᶜ	Ṭ	ṭ	Ů	ů
Ǔ	ǔ	Ẓ	ẓ						

The next step consists in the creation of files that will be used later in the system training. The files' construction method can vary according to the characters position and the appearance frequency. This variation influences directly on the quality of the system recognition and its success rate.

V. Proposed System

To deal with the Amazigh language transcribed into Latin letters with diacritical marks, we propose an OCR system based on the nonlinear binarization method in the preprocessing phase, in addition to a structural approach based on polygonal approximation features for the classification phase.

A. System architecture

The proposed system architecture is as follow:

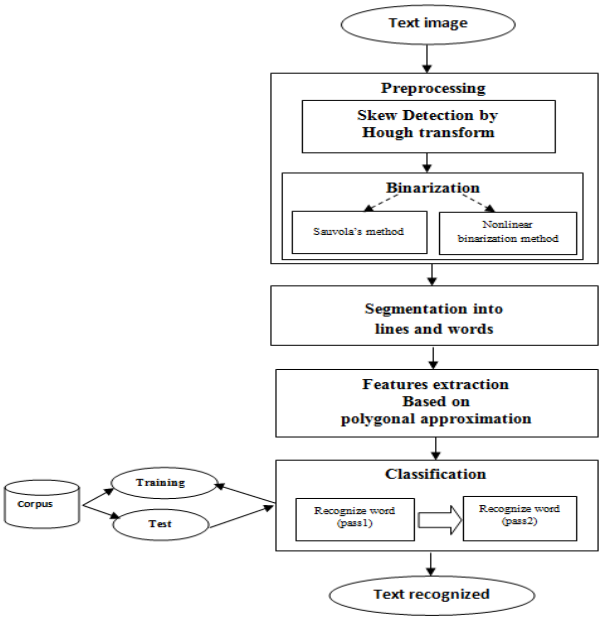


Fig. 3 The architecture of our system

Text image is entered to the system after being scanned by a scanner. Then, the image undergoes some preprocessing in order to increase image's quality.

The first treatment is the skew detection and correction. For this purpose, we use the Hough transform.

Hough transform is a method that allows detecting lines in an image. It can be applied to any geometric shape that can be described by equation (1)

$$\rho = x*\cos\theta + y*\sin\theta \quad (1)$$

Where:

(ρ,θ) defines a vector from the origin to the nearest point on the line.

For the binarization phase, we try two methods: the first is the nonlinear binarization method [17], which is a compute-intensive method. It works well for degraded and historical book pages. The 2nd method is Sauvola, based on computing the threshold using the dynamic range of image gray-value standard deviation [18].

In the features extraction phase, we use the polygonal approximation fragments as features.

The polygonal approximation can be obtained by choosing the polygon vertices in such a way that the overall approximation error is minimized.

Error measures are defined in (2) and (3).

$$\text{Mean square} \quad E_2 = \sum_{i=2}^{N-1} |x_i - d_i|^2 \qquad (2)$$

$$\text{Maximal} \quad E_{max} = \max_{2 \le i \le N-1} |x_i - d_i| \qquad (3)$$

Then, we attempt a first pass classification to recognize each word in turn. Each satisfactory word is passed to an adaptive classifier as training data. The adaptive classifier receives recursively new training data to enhance learning and allow the recognition of others words in the second pass.

To implement the features extraction and the classification phases, we use the Tesseract recognition tool.

B. Tesseract tool

Tesseract is an optical character recognition engine for various operating systems. It was originally developed at HP between 1984 and 1994 [19], [20]. It was modified and improved in 1995 with greater accuracy. In late 2005, HP released Tesseract for open source. Now, it is developed and maintained by Google.

The first relevant criterion in Tesseract is the fact that is free and open source (FOSS), which is an advantage and a key point in the research development.

Usually, whenever Tesseract is compared to another free OCR tool, it is the best whether in terms of recognition rate or speed [21]. Even, when it is compared with the Finereader [22] commercial tool. Tesseract arrives to rub it and managed to overtake for handwritten writing [23].

The specificity of the Amazigh language transcribed into Latin characters is the presence of diacritic below and above a large number of characters. The experiments on Tesseract for diacritical languages, such as ancient Greek [20] and Urdu [24], have shown that it is strong enough for this type of languages.

Hence, the interest to train this tool on Amazigh language transcribed on Latin characters. The process of training passes by tree steps: generation of corpus, creation of the traineddata file and the training [19], [25].

VI. Experiments & Results

The aim of this section is to evaluate and improve our system, by making tree different experiences. The first experience concerns the training corpus composition. In this experience, we vary two criteria: the corpus composition and the characters size. The second experience is about the preprocessing phase. The third experience is based on the evaluation of the proposed system.

To perform these tests, we use a collection of documents. Part of this collection has undergone a pre-treatment to increase the image quality, while the other part is kept with low quality, in order to view the system behavior in both cases.

Fig 4 and Fig 5 show an example of those documents.

Lqaɛida n ugdal, ar as ttbṛṛmnt tikrkurin baš ad ssnn mddn is illa ugdal. Ass nna ira irẓm ugdal, ar ittili lbriḥ, ar ttinin: "Hann ag^{w}dal irẓm!"

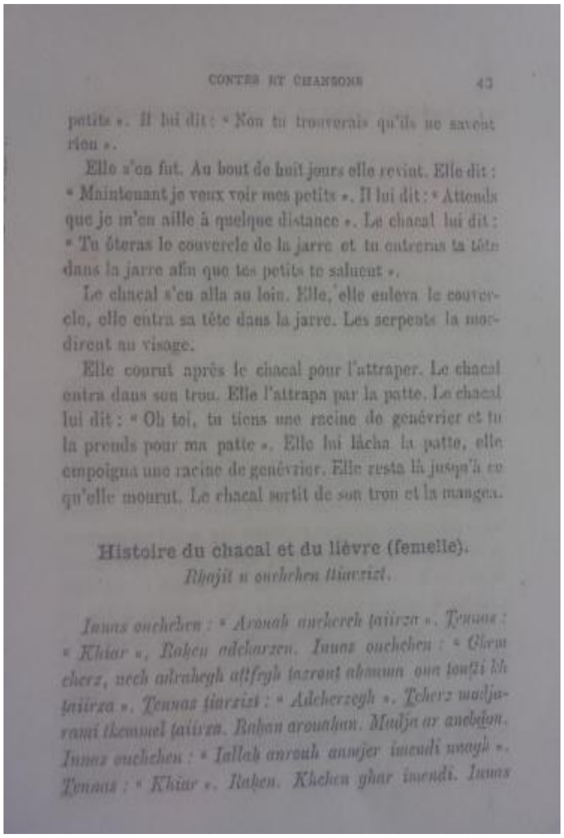
Ar fttunt tmγarin ula lḥšum ula irgazn, ar grrun afiyyaš; ar t id ttggan γ taryilin, γ a ttgrrun ar t id ttasin ar yan lmudɛ iɛdln, ffin t gis, skrn gis agudi; ar ass nna kullu t g^{w}ran, ar ttggan ifrig n ugudi ad ur lkmnt lbahim. Wan dar ur illi wargan, ar igrru γ aylli nn iqaman γ ssjrt nγ γ ddu ssjrt.

Targant nna illan γ usulil ur stt ttḥkamn mddn a stt grun, ar srs aqqlaynt lbahim ar stt grrunt; iγ ddant lbahim ar d ulsunt γ tuẓẓumt n uzal ar sgluliynt uẓlim lli illan γ udis nnsnt. Uẓlim ann lli sgluliynt ar t smunan, ar t id ttɛzaln waḥdut, ašku illa gis wargan ifulkin, yuf walli yaḍni šfiyyašn mddn, ar gis ttili tujjut iɛdln iqwa bahra; γikann a f a t id ttɛzaln.

Ar d ttawin mddn lbahim s dar ugudi n ufiyyaš lli illan γ tagant, ar t id ttɛmmarn γ išwariyn, asin t f iggi lbahim, ar t id ttawin s tgmmi, ar t inn gis srusn. Ass nna kmmln s usatti, ar t sfiyšn s uẓṛu, ɛzln gis alig s yat tsga, uẓlim s yat tsga. Alig ar t sttan izgarn ula irɛman; uẓlim ar t ttrragn s uẓṛu. Iγ idda ar t rgin, ar grrun tiẓnin nns γ tuẓẓumt n yirgn[23]; irgn ar tn ttggan i lɛfiyt; tiẓnin n

20. Fruit vert de l'arganier. (Laoust)
21. Fruit mûr. (Laoust)
22. Fruit mûr qui tombe. (Laoust)
23. Noyau cassé. (Laoust)

Fig. 4 An example of good quality document



CONTES ET CHANSONS 43

petits ». Il lui dit : « Non tu trouverais qu'ils ne savent rien ».

Elle s'en fut. Au bout de huit jours elle revint. Elle dit : « Maintenant je veux voir mes petits ». Il lui dit : « Attends que je m'en aille à quelque distance ». Le chacal lui dit : « Tu ôteras le couvercle de la jarre et tu entreras ta tête dans la jarre afin que tes petits te saluent ».

Le chacal s'en alla au loin. Elle, elle enleva le couvercle, elle entra sa tête dans la jarre. Les serpents la mordirent au visage.

Elle courut après le chacal pour l'attraper. Le chacal entra dans son trou. Elle l'attrapa par la patte. Le chacal lui dit : « Oh toi, tu tiens une racine de genévrier et tu la prends pour ma patte ». Elle lui lâcha la patte, elle empoigna une racine de genévrier. Elle resta là jusqu'à ce qu'elle mourut. Le chacal sortit de son trou et la mangea.

Histoire du chacal et du lièvre (femelle).

Rhajit n ouchchen tiiarzizt.

Innas ouchchen : « Arouah ancherch taiirza ». Tennas : « Khiar », Rahen adcharzen. Innas ouchchen : « Ghem cherz, nech adraheg attfegh tazrout ahaoun oun toufti kh taiirza ». Tennas tiarzizt : « Adcherzegh ». Tcherz madjarant tkemmel taiirza. Rahan arouahan. Madja ar anebdou. Innas ouchchen : « Iallah anrouh anmjer imendi unagh ». Tennas : « Khiar ». Rahen. Khchen ghar imendi. Innas

Fig. 5 An example of bad quality document

A. Corpus composition

As explained in Section 4, the training corpus can be built in different ways, depending on the character position, the characters' appearance frequency and size.

In this work, we tried to build our corpus in two different ways and test the effect of this change on the recognition results.

Corpus1: this corpus is organized as follows: characters are classified according to alphabetical order, uppercase followed by lowercase, separated by a space. Non-letters characters (punctuation, parenthesis, accolades, ...) are placed at the end of the set. The overall characters are repeated 10 times so the appearance frequency is the same for each character.

Corpus 2: it is built based on a text extracted from a book written in the studied language. It contains all character including uppercase, lowercase and non-letters characters. The appearance frequency is different, high for frequently used character, such as "ṡ" and "ẕ", and equal to 5 for rarely used character, such as "Ŭ" and "Ā".

For both corpora, we varied characters size from 9pt to 96pt in order to determine the appropriate size.

Tests are performed on a good quality document. The results obtained are shown in the following table:

Table II Recognition rate according to the training corpus and the font size

Size (pt)	Corpus 1	Corpus 2
9	2%	1%
10	5%	6%
11	1%	7%
12	15%	16%
13	16%	18%
14	21%	28%
18	42%	51%
24	77%	60%
36	82%	92%
42	79%	86%
64	75%	74%
72	65%	66%
96	65%	63%

We note that the recognition rates vary according to the corpus type and size.

For both corpora, the size of 36pt is the best. However, the training corpus 2 gives better results against corpus 1 in most cases. Therefore, the appropriate training corpus to use is the corpus 2 with the font size of 36pt.

B. Preprocessing variation

Pretreatment is an important phase in the OCR system. It aims to improve the quality of the image and consequently increase the rate of recognition.

The problem of the writing inclination is generally due to the acquisition phase. To correct this problem, we use the Hough transform. The results are shown in the following figure:

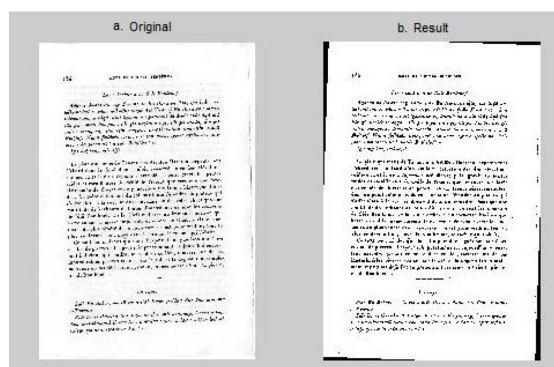
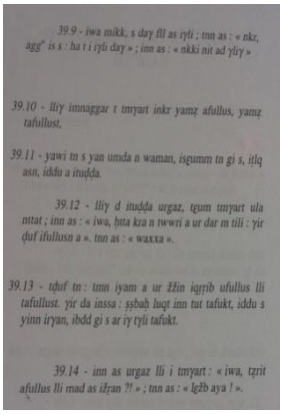


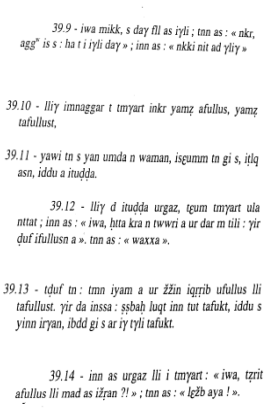
Fig. 6 Skew detection and correction of a document

As mentioned above, we tried two binarization methods: the nonlinear binarization and Sauvola methods.

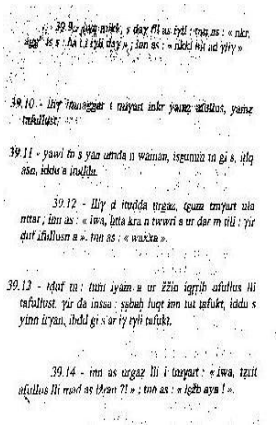
The impact of these methods is shown in Fig 7 and table III.



a. Original image



b. Nonlinear binarization method



c. Sauvola's method

Fig. 7 Binarization with different methods: a-Original image b-Nonlinear method c-Sauvola's method.

Table III Recognition rate according to the binarization method

Image	Original	With nonlinear binarization method	With sauvola binarization method
Recognition rate	75%	89%	82%

The figure and table above show that the recognition with pretreatment gives better results compared to the original document.

The comparison of the two methods used for binarization reveals that the nonlinear binarization method gives remarkable results visualized at the figure and approved by recognition rate in table III.

C. Approach test

To evaluate our system, we use a set of document, containing 220 pages collected from 4 different books [16], [26], [27], [28], written in Amazigh language transcribed into Latin.

Furthermore, we use two metrics:

- The recognition rate that achieves 89%, according to the precedent experiences;
- The confusion matrix that evaluates the system performance by calculating the number of confusion errors between character classes.

As mentioned before, the character set used for writing the Amazigh language transcribed in Latin is composed of Latin characters with diacritical marks.

To carry on the evaluation of our system, based on the confusion matrix, we have restricted the matrix to the diacritical characters, given the high number of studied characters (115), especially, that research on diacritical-free Latin alphabets has known a success, and the recognition rates based on polygonal approximation feathers are high for these characters [10].

We note that this test is performed on a document composed of 200 characters for each class.

Table IV Confusion Matrix between characters

	ā	ă	ḅ	ḍ	ĕ	ġ	ḡ	ḥ	ḫ	ḷ	ŏ	ṛ	ṡ	š	ṭ	ů	ŭ	ẓ	ɛ	ɣ
ā	200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ă	4	196	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ḅ	0	0	200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ḍ	0	0	0	200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ĕ	1	2	0	0	197	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ġ	0	0	0	0	0	197	3	0	0	0	0	0	0	0	0	0	0	0	0	0
ḡ	0	0	0	0	0	4	196	0	0	0	0	0	0	0	0	0	0	0	0	0
ḥ	0	0	1	0	0	0	0	199	0	0	0	0	0	0	0	0	0	0	0	0
ḫ	0	0	0	0	0	0	0	5	169	0	0	0	0	0	0	0	0	0	0	0
ḷ	0	0	0	0	0	0	0	0	0	195	0	3	0	0	2	0	0	0	0	0
ŏ	0	2	0	0	0	0	0	0	0	0	198	0	0	0	0	0	0	0	0	0
ṛ	0	0	0	0	0	0	0	0	0	4	0	192	0	0	2	0	0	2	0	0
ṡ	0	0	0	0	0	0	0	0	0	0	0	0	195	5	0	0	0	0	0	0
š	0	0	0	0	0	0	0	0	0	0	0	0	7	193	0	0	0	0	0	0
ṭ	0	0	0	0	0	0	0	0	0	4	0	3	0	0	193	0	0	0	0	0
ů	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	190	10	0	0	0
ŭ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	191	0	0	0
ẓ	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	192	0	0
ɛ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	200	0
ɣ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	200

From this experimentation, we remark that misclassification errors are detected, especially for diacritical characters and its body, such as "ṭ" that is confused with "t", "ḍ" with "d" and "ḥ" with "h".

VII. Conclusion

In this paper, we discuss the optical character recognition of documents, which is an active area of research today. We have introduced the OCR system and its modules. Then, we have presented the state of the art of the approaches developed for each module. We have described our proposed OCR system architecture for the Amazigh language transcribed into Latin, and the steps to construct an associated OCR corpus, in order to train and test the system.

In the experimentation, we performed a set of binarization techniques, in the pretreatment phase, in order to define the most appropriate method for our documents. We used a structural approach based on the polygonal approximation in the classification phase and we have reached a recognition rate of 89%.

This study opens several perspectives such as improving the preprocessing phase by applying other treatments to increase the recognition rate, and enriching the used corpus. This enrichment consists on adding a new set of characters and executing the training for new fonts.

REFERENCES

[1] L. Eikvil., "OCR, Optical Character Recognition", Norsk Regnesentral, 1993

[2] A. Belaïd, "Reconnaissance automatique de l'écriture et du document", Campus scientifique, Vandoeuvre-Lès-nancy, 2001

[3] R. El Ayachi, M. Fakir, B. Bouikhalene, "Recognition of Tifinaghe Characters Using Dynamic Programming & Neural Network", Recent Advances in Document Recognition and Understanding, 2011

[4] N. Aharrane, K. EL Moutaouaki, Kh. Satori, "Recongnition of handwritten Amazigh characters based on zoning methods and MLP", Wesas transactions on computer, volume 14, 2015

[5] P. Charles, V. Harish, M. Swathi, CH. Deepthi, "A Review on the Various Techniques used for Optical Character Recognition", International Journal of Engineering Research and Applications, 2012
[6] Z. Darko, P. Janez, D. Andrej, "Document Categorization Based On OCR Technology: An Overview", Proceedings of the 7th European Computing Conference (ECC '13) Dubrovnik, Croatia June 25-27, 2013
[7] N. Noor, "Bangla optical character recognition", Thesis, 2005
[8] R. Bousslimi, "Système de reconnaissance hors-ligne des mots manuscrits arabe pour multi-scripteurs", Mémoire de mastère 2006
[9] I. Chaker, R. Benslimane, "Nouvelle approche pour la reconnaissance des caractères arabes imprimés", Revue Méditerranéenne des Télécommunications, VOL.1 No.2, 2011
[10] A. Muaz, "Urdu Optical Character Recognition System", Thesis, 2010
[11] Ja Palkan., Ji Palka., M. Navratil, "OCR systems in language specific environments", the 10th WSEAS International Conference on Telecommunications And Informatics (TELE-INFO '11), Canary Islands, Spain, May 27-29, 2011
[12] P. Singh, S. Budhiraja, "Feature Extraction and Classification Techniques in O.C.R. Systems for Handwritten Gurmukhi Script – A Survey", International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 Vol. 1, Issue 4, pp. 1736-1739
[13] A. Roux , "Choix de Version Berbères Parler du Sud-Ouest Marocaine", France, 1951
[14] A. Skounti, A. Lemjidi, M. Nami, "Tirra aux origines de l'écriture au Maroc, Publications de l'Institut Royal de la Culture Amazigh, Rabat", 2003
[15] E. Laoust, "Mots et Choses Berbères", Paris, 1920
[16] H. Stroomer, "The argan tree and its tashelhiyt berber lexicon", Université de Leyde, Etudes et document berbères, 2008
[17] N. Ntogas, D. Veintzas, "A binarization algorithm for historical manuscripts", Presented at the Proc. 12th WSEAS Internat. Conf. on Communications, Heraklion, Greece, pp. 41–51, 2008
[18] K. Khurshid, I. Siddiqi, C. Faure, N. Vincent, "Comparison of niblack inspired binarization methods for ancient documents", Proc. 16th International Conference on Document Recognition and Retrieval, USA, 2010
[19] S. Ray, "An Overview of the Tesseract OCR Engine", International Conference on Document Analysis and Recognition, 2007
[20] W. Nick, "Training Tesseract for Ancient Greek OCR", Google Inc "eutypon28-29", October 2012
[21] S. Dhiman, A. Singh, "Tesseract Vs Gocr A Comparative Study", International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-2, Issue-4, September 2013
[22] http://Finereader.abbyy.com
[23] M. Heliński, M. Kmieciak, T. Parkoła, "Report on the comparison of Tesseract and ABBYY FineReader OCR engines", Improoving Access to Text.
[24] A. Qurat ul Ain, H. Sarmad, N. Aneeta, A. Umair, I. Faheem, "Adapting Tesseract for Complex Scripts: An Example for Urdu Nastalique", 11th IAPR Workshop on Document Analysis Systems (DAS 14) 2014
[25] http://code.google.com/p/tesseract-ocr
[26] C. Justinard, "Anuel De Berbere Marocain (Dialecte Rifain)", Librairie Paul Geuthner, Paris 1926
[27] B. Lasri Amazigh, "Ijawwan N Tayri", Marrakech, Imp Imal, 2008
[28] A. Leguil, "Conte Bereber Grivois Du Haut Atlas", L'Harmattan, Paris 2000