ISSN 2007-9737

Neural-Combinatorial Classifiers for Arabic Decomposable Word Recognition

Zeineb Zouaoui*, Imen Ben-Cheikh, Mohamed Jemni

University of Tunis,
National High School of Engineering of Tunis,
Department of Computer Science,
Tunisia

{zeineb.zouaoui, imen.becheikh}@gmail.com, mohamed.jemni@alecso.org.tn

Abstract. Recognition tools and techniques for Arabic script are still under development due to the topological ambiguities and inflectional nature of this language. In this regard, this paper presents an approach based on a combinatorial optimization technique incorporating convolutional neural networks for Arabic word recognition. We handle a wide vocabulary of Arabic decomposable words. We adopt a design that resembles a molecular cloud with words structured according to their roots and patterns. This conception fits well with the Arabic linguistic philosophy of building words from their roots. Hence, each sub-vocabulary represents a sub-cloud, encompassing neighboring words derived from the same root and following different patterns and forms of derivation, inflection and agglutination (proclitic and enclitic). Hence, each sub-vocabulary represents a sub-cloud, encompassing neighboring words derived from the same root and following different schemes and forms of derivation, inflection and agglutination (proclitic and enclitic). Accordingly, as a first step, we have used a recognition approach based on the metaheuristic method of simulated annealing (SA). In a second work, we implemented the SA algorithm by integrating linguistic knowledge. Extending this work, we choose to integrate a convolutional neural network into the recognition process of the SA algorithm to benefit from the advantages of both methods. To conduct our experiments, which yielded promising results, we use a corpus of Arabic words including samples and agglutinated words from the APTI database.

Keywords. Convolutional neuronal network, combinatorial optimization, simulated annealing, morphological characteristics, Levenshtein distance.

1 Introduction

Text recognition remains a very active area of research. Which aims to read texts efficiently, similarly to human reading process. The recognition process is challenging mainly due to the adopted approach (structural, analytic), script language and vocabulary size. These ongoing challenges handling with Arabic language are due to its cursive nature, script types and morphological peculiarities.

Vocabulary size (reduced or large) also influences the performance of a recognition system as well as the adopted method (stochastic, neural, etc). Although there are several approach classes for characterizing and recognizing Arabic scripts, the obtained results did not yield performances similar to those achieved for other scripts, like Latin.

To face these challenges, a robust methodology becomes necessary. Hence, they are two radically different approaches, machine learning and combinatorial optimization can be used together to serve a common goal and solve the same problem [1].

On one hand, training algorithms are able to learn and generalize on the basis of unstructured or unformalized information. On the other hand, combinatorial optimization methods are frequently proposed to solve pragmatically artificial intelligence problems with large solution spaces. We attempt then to hybridize these two types of approaches in order to develop new methods of

ISSN 2007-9737

Fig. 1. Samples of different shapes of the same letter (ع) **/E/** in different positions

resolution and novel perspectives by taking into account the major challenges associated with the complexity and specificity of Arabic scripts.

The primary contributions of this paper can be summarized as follows: (1) To focus on incorporating linguistic knowledge into the recognition process, thanks to the importance of lexical and syntactic information in written words. (2) To benefit from combinatorial optimization and machine learning methods in order to propose an approach that handle with large vocabulary of Arabic decomposable words (derived from roots) presenting various morphological features (derivational, inflectional and agglutinative).

The remainder of this paper is structured as follows: Section 2 presents a short morphological and topological analysis of Arabic script. Section 3 describes natural language processing for Arabic script recognition.

Section 4 displays a brief description of combinatorial optimization techniques. A convolutional neural network is illustrated in section 5. Section 6 reviews the related works suggested in the literature. Section 7 introduces our proposed approach. Then, section 8 reports our experiments by focusing on the major findings. Finally, section 9 summarizes the paper and highlights major directions for future research.

2 Arabic Script Analysis

In this section, we analyses Arabic words by taking into account two major aspects, namely the linguistic aspect and the morphological aspect.

2.1 Topological Peculiarities

- The Arabic language has a huge vocabulary, letters are mostly connected, and most words consist of more than one sub-word, called PAW (Pieces of Arabic Word).
- An Arabic script is characterized by the presence of a horizontal baseline or a reference line. This latter is the place of horizontal ligature characters of the same string.
- Arabic characters are written cursively from right to left in both manuscript and print.
- The Arabic alphabet comprises 28 basic characters, 16 of these characters include one, two, or three diacritical dots in their various forms.
- These dots distinguish between characters of the same body. They can also be located above or below the body of the basic character.
- An Arabic letter has different forms according to their positions in the word: at the beginning, in the middle or at the end of the word (see Fig.1).

These particularities distinguish Arabic from other languages. Moreover, their absence can cause some problems. For instance, the absence of diacritical dots and vowels causes confusion between letters of the same shape. Fig.2 illustrates samples of these problems.

The word "جمل" **/jamal/**[1] with diacritics that means a camel -differs from the word "حمل" **/Hamal/** without diacritics- that means a lamb. Moreover, an Arabic text without vowels is highly ambiguous and the correct functions of words cannot be distinguished.

For example, the word "علم" **/Elm/** (without voyellation) is ambiguous as it has different meanings depending on its voyellation. It can mean science "عِلْمٌ" **/EilomN/**, flag "عَلَمٌ" **/EalamN/**, the past tense verb (he knew) "عَلِمَ" **/Ealima/** or "عُلِمَ" **/Eulima/** (has been known).

Taking another example, the absence of capital letters in the Arabic word "جميلة" **/jamiylap/** can generate ambiguity in determining its function: whether an adjective, a proper noun? It can mean either the adjective "beautiful" or the proper noun "Jamila".

[1] Transliteration is coded following Buckwalter transliteration: https://www.ipabwat.com/

ISSN 2007-9737

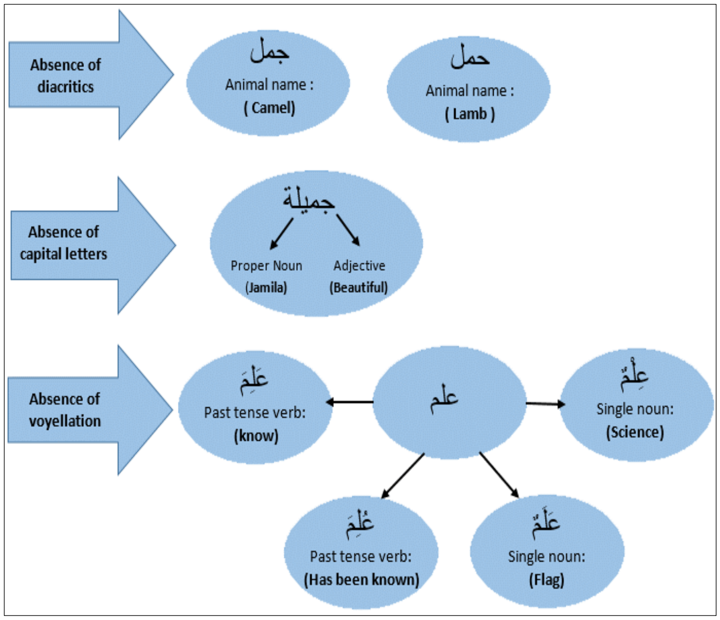

Fig. 2. Examples of ambiguities related to the absence of diacritics, voyellation and capital letters in arabic words

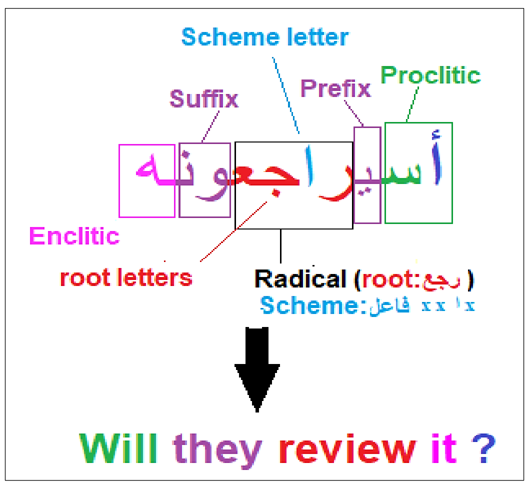

Fig. 3. The agglutinative derivation of the single word "أسيراجعونه" **/>asayuraAjiEuwnahu/**

2.2 Morphological Peculiarities

Morphological analysis is a challenging task in Arabic natural language processing compared to other languages. This is due to the diversity and complexity of words linguistic characteristics.

For instance, (see Fig.3), a full English sentence can be translated into a single Arabic word; such as the sentence **"Will they review it?"** that can be replaced by the word "**أسيراجعونه**" **/>asayuraAjiEuwnahu/**. Furthermore, an Arabic word is either decomposable or non-decomposable. Non-decomposable words cannot be divided into affixes (prefixes, infixes and suffixes) and roots. This word category includes pronouns, proper nouns, numbers, country names, and particles (equivalent to adverbs, prepositions, conjunctions, etc.).

doi: 10.13053/CyS-28-3-4368

ISSN 2007-9737

Various non-decomposable words are made up of new non-Arabic words that have found their way into the language by taking into account some terms associated with modern technology [2].

A decomposable word, however, results from its root derivation according to a conjugated scheme. A root is purely consonantal; that is to say it is formed by adding a sequence of three, four, or even five consonants (for nouns) in order to build the base of the word. A root is a salient element in derivative languages. Indeed, each root corresponds to a semantic field.

Thanks to different schemes, we can generate a family of words in each semantic field. The scheme is composed of three different consonants **ف** /f/, ع **/E/**, and ل **/l/**, which are vocalized and can be augmented by adding other letters, namely prefix, suffix, and infix. The root derivation gives a word its radical meaning according to the short scheme as explained in Fig.3.

This paradigm allows the generation of new words by adding prefixes that precede a root base, by injecting infixes that appear between the basic letters and by adding suffixes coming at the end. This morphological diversity is also clear in the inflectional process. In fact, a radical undergoes inflected conjugations that are able to change the prefixes and suffixes and/or adjoin additional letters. In this way, the elements of inflectional conjugations of Arabic comprise the personal pronoun, time, gender, number, function (accusative or nominative), etc.

Arabic language morphology is also characterized by its agglutinative aspect. Agglutination can be divided into two main kinds: proclitics and enclitics [3]. A proclitic can be either simple morphemes of one letter (coordinators, conjunctions, and prepositions) or complex morphemes of multiple letters (the combination of simple proclitics).

The enclitic is a supplement pronoun that can be single or double and is linked to the word that precedes it. Fig.3. describes the derivation process of the agglutinated word **سيراجعونه** /**sayuraAjiEuwnahu**/. Which results from the conjugation of the root **رجع** **/rajaEa/** according to the scheme **فاعل** **/faAEala/** (x x ا x; where x refers to one of the three consonants) with the pronoun **هم** **/hum/** (they: plural masculine) by adding both the proclitic **س** **/s/** (designating the future tense) and the enclitic **ه** **/h/** (designating the object complement: him).

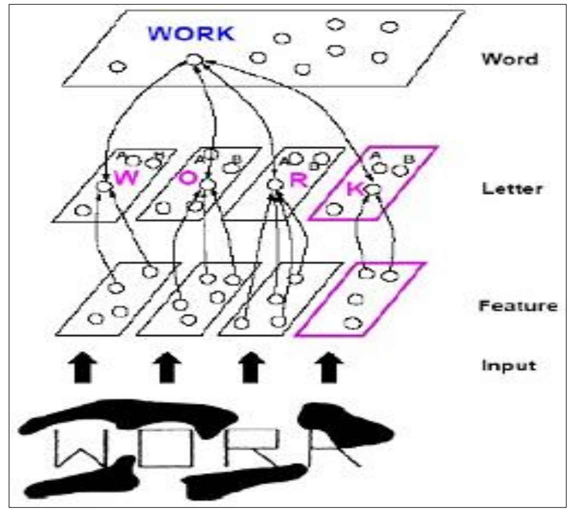

Fig. 4. A perceptual model inspired by human vision system [8]

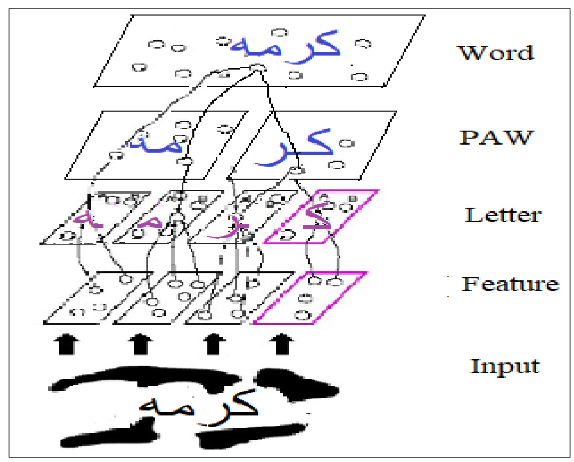

Fig. 5. A perceptual model influenced by the word superiority effect

3 Natural Language Processing

Pattern recognition is the automation of artificial perception tasks performed by the human sensory system and brain. It aims to classify entities into categories on the basis of observations made on them [4].

ISSN 2007-9737

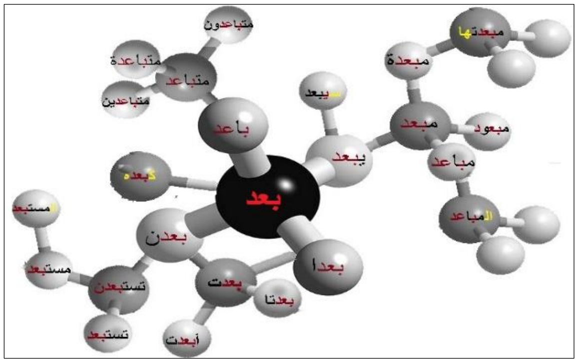

Fig. 6 Parts of the sub-cloud corresponding to the root "بعد" **/baEada/** (i.e. to go far)

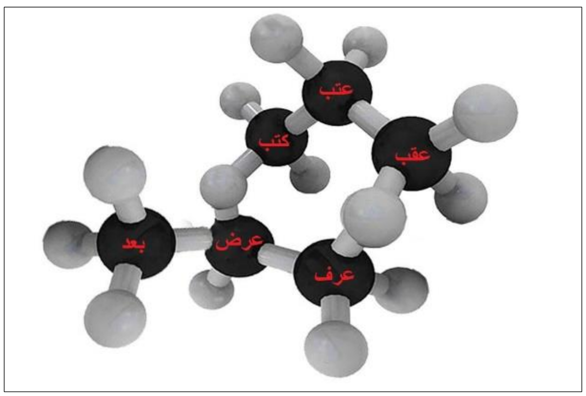

Fig. 7 CNN-Molecular cloud: the association of several sub-clouds (black balls symbolize sub-cloud cores corresponding to the roots

In this respect, psychoanalytic studies show that humans memorize letters standing for the whole word instead of sequences of separate letters. Therefore, letters of words can be memorized better than letters in meaningless sequences. The "Word Superiority Effect" concept, proposed by Mc Clelland and Rumelhart [5, 6], was inspired by this human reading perception (see Fig.4). It permits a layered representation of the word, from the local to the global layer and viceversa.

According to [6], this model is applicable to Arabic script recognition on condition that an intermediary global level, known as the level of the pseudo word (PAW), is added as shown in Fig.5. Therefore, word recognition can be guided by the meaning of different possibilities of PAW combinations.

The importance of the "Pseudo-Word Superiority Effect" decline not only from the primary function of PAWs in the Arabic language, but also from the analogy with the "Word Superiority Effect", since PAWs lead to not totally cursive writing.

Indeed, some Arabic letters change their shapes to maintain cursive continuity of the writing, where others maintain their shape including space between letters and introducing, thus, non-cursiveness into writing. In the same direction, the integration of linguistic information in the recognition process is one of the most promising research approaches.

In this context, studies in references [3,7], which focused on the "Word Derivation Effect" by integrating linguistic information such as roots and patterns in the Arabic word recognition process, yielded some interesting results. Accordingly, we generally estimate that word recognition can occur at several independent, but complementary, levels, such as root recognition, pattern recognition, agglutination recognition, recognition conjugate elements, etc.

Then, we assume that the human reading process is also characterized by the notion of "Word Morphology Effect". To substantiate this, we will prove that the incorporation of morphological knowledge in a perceptual model helps us to recognize a large vocabulary of Arabic words. To this end, the suggested approach combines Arabic morphology with the human reading process.

4 Combinatorial Optimization

Combinatorial optimization (CO) problems consist in choosing the best solution out of a very large, but finite, set of solutions. Hence, solving a CO problem means finding an optimal function from a very large but finite number of choices. To solve CO problems, it is necessary to choose the most appropriate method, (either exact or heuristic method).

Exact methods explore the whole search space in order to find at least one of the optimal solutions for a given problem. These methods have mostly been used on small problem instances. On the contrary, heuristic methods explore a subset of the search space in order to find a practical solution

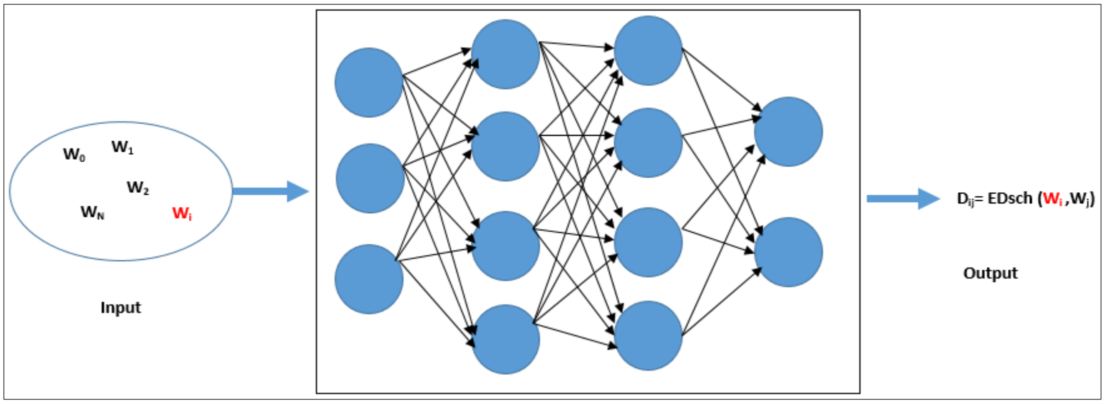

Fig. 8 CNN architecture

within a reasonable time. They can be often applied to all types of problems. Moreover, heuristic methods are divided into two sub-classes, namely meta-heuristic approaches and constructive approaches.

The former search for the global optimum over a set of local optimums, while the latter, also known as iterative methods, build a complete solution step by step. Metaheuristics fall into two major categories: evolutionary algorithms (e.g., genetic algorithms, etc.) and neighborhood methods (e.g., Tabu Search, Simulated Annealing, Hill Climbing, etc). The former deal with sets of solutions at the same time to quickly find the optimal solution.

The latter, however, start their search from a simple initial solution that will be modified and improved through research in order to achieve an optimal solution.

Finally, the choice of an appropriate combinatorial method depends on some characteristics, such as the problem complexity, cost evaluation, the existence of constraints, the desired results (the number of solutions and their properties), etc.

This information allows us to select the best technique to be used from the numerous available alternatives. Despite their differences, combinatorial optimization and machine learning methods aim to reach a common goal and solve the same problem. The hybridization of combinatorial optimization and learning methods leads to new methods of resolution and novel perspectives [1]. In this regard, hybridization consists in fully exploiting each method's strengths in order to obtain the best performing solution.

5 Convolutional Neural Network

A convolutional neural network (CNN), originally proposed by LeCun [9], is a neural network model. It combines three key architectural ideas: local receptive fields, shared weights, and spatial subsampling. CNN architecture is composed of three main layers:

- Convolutional layer applies a filter to an input image. The convolution value is calculated by taking the scalar product of the corresponding kernel values and the channel matrices.
- Pooling layer is generally placed after the convolutional layer. Its main utility consists in reducing the spatial dimensions (width/height) of the input volume for the next convolutional layer. In the pooling layer, we use a sliding window moving over the input in large steps. With each movement of the window, representative values of data involved in the window are detected.
- Max pooling takes the maximum value in each window. It is preferred over other techniques thanks to its performance characteristics.
- Fully connected layers are fully connected to the previous output layer. They are commonly used at the end of a CNN to connect each hidden layer to the output layer and then build the required number of outputs.

```
W = select a random word from the cloud (current solution)
EDsch (W) = New Edit distance between the unknown Word and the current solution W
Set the initial temperature initial_T
While (temperature >0 and non-convergence)
        Neighbor = select the best neighborhood word solution
        Calculate delta = EDsch (neighbor) – EDsch (W)
        If (delta <= 0)
            W = neighbor
        Else
            Select new neighbor with probability e^-(delta/t)
        End If
        Decrease the temperature
End While
Output the final solution
```

Fig. 9. A simulated annealing pseudo algorithm with a new Edit distance [20]

Table 1. Correspondence between a physical system and an optimization problem

Optimization problem	Physical problem
Solution	System state
Objective function	Free energy (E)
Problem parameters	Coordinates of particles
Find a good configuration	Find low energy states
Global optimum	Ordered stable state
Local optimum	Metastable state
Local search	Rapid quenching
The T parameter	Temperature

Table 2. Examples of classic vs new edit distance calculation on words following different schemes (sch)

Input		Classic distance ED	New distance EDsch
يبعد (sch= فعل)	يبتعد (sch=افتعل)	1	2.9
يبعد (sch= فعل)	مبعد (sch= مفعل)	1	2.6
يبعد (sch= فعل)	يبعده (sch= فعل)	1	1.3

The convolutional neural network has various assets. First, feature extraction and classification are integrated into a single structure that is fully scalable.

Second, the network extracts 2D images at multiple dyadic scales. Third, it is relatively invariant to local geometric distortions in the image.

CNN has been used in several applications, such as digit number recognition, face detection and facial recognition.

6 Related Works (Linguistic and/or CNN and/or Combinatorial)

Many experiments have shown that human readers, unlike automatic recognitions software, are skilled in combining syntactic, semantic, and morphological analyses. In reading Arabic written expressions, humans rely on the very regular word-form structure. Following this word perception, [10] set forth an affixal approach for Arabic script recognition.

It consists in segmenting words into letters and recognizing their morphological entities. It is an analytical approach to segmentation, which allow authors to integrate linguistic knowledge into the coherence verification of prefixes and affixes, rather than in the recognition process.

In addition, an “affixal approach” was proposed in [2, 7, 3]. Unlike [10], which first recognize letters locally and then words, Markovian and neural classifiers in [2, 7, 3] were suggested in order to first recognize linguistic word concepts globally (prefixes, roots, schemes, etc.).

For instance, [11] put forward two systems for Arabic offline handwriting recognition. The first system uses the connectionist temporal classification (CTC) combined with Bidirectional Long Short-Term Memory (BLSTM) architecture.

The second system, however, is based on cascade CNN and MDLSTM layers. The experiments were carried out on the KHATT database and proved the performance of the two proposed systems.

Reference [12] offered a spotting system for printed and handwritten text line images. It used a hybrid architecture composed of a deep bidirectional neural network, long-term memory and a hidden Markov model (HMM). It exploits not only the robust learning capability of deep neural network representations, but also HMMs sequential modeling ability.

Their system was tested on two Arabic databases (KHATT, PKHATT) and the RIMES Latin database. The experimental results on script identification and keyword extraction confirm the performance of the suggested approach. In a study conducted by [13], a new model was proposed based on deep neural networks for offline Arabic handwritten text recognition (digits, characters and single words).

In fact, the author proposed a supervised convolutional neural network (CNN) model that contextually extracts optimal features and employs batch normalization and dropout regularization parameters.

The model was tested and proven on the MNIST English Isolated Figures database with an accuracy of 99.94% and a precision of 99.68%.

Some researchers have recently attempted to benefit from the advantages of automatic learning in order to solve combinatorial optimization problems. In this context, we can cite the research work of [1], which utilized learning and combinatorial optimization techniques for graph matching, pattern recognition and classification, and computer vision problems.

Moreover, [14] recommended an approach to solve combinatorial optimization problems using neural networks (LSTM) and reinforcement learning. They focused on the traveling salesman problem and trained a recurrent neural network that, given a set of city coordinates, predicts a distribution over different cities permutation.

This approach achieves optimal results on 2D Euclidean graphs with up to 100 nodes.

7 Proposal of the Approach

The major objective of this work is to propose an approach for recognizing a large vocabulary of Arabic words, and exploiting the benefits of neural methods and combinatorial optimization techniques. These mechanisms consist of three different steps: In the first step, the vocabulary

ISSN 2007-9737

Fig. 10. Samples of APTI database

(a)

(b)

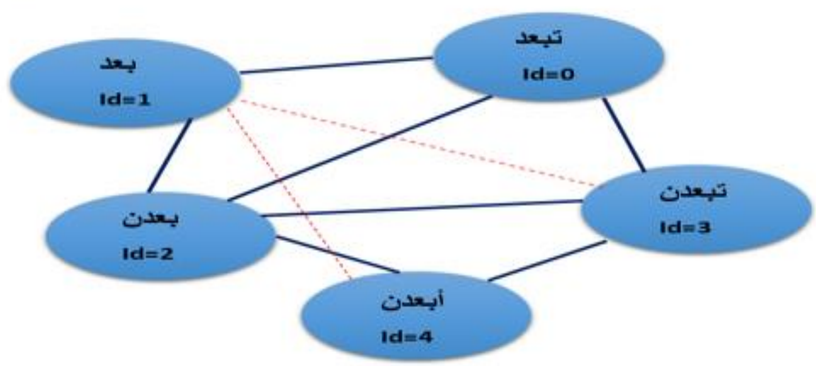

Fig. 11 Sample of neighbor's corpus: (a) Snippet of word corpus (b) Snippet of word neighborhood

Table 3. Some word CNN samples

	Arabic Word	Transliteration	Score
CNN_0/0/6/10/4/0	تبعدن	taboEudona	99,48%
CNN_0/0/5/11/5/0	بعدن	baEadona	99,23%
CNN_0/2/5/10/0/0	باعدتما	baAEadotumaA	88,52%

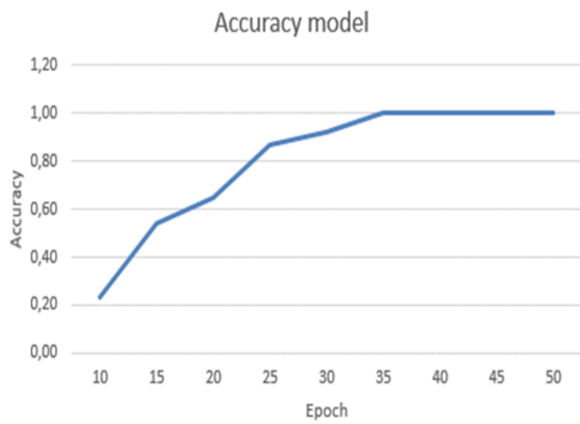

Fig. 12. Evolution of training data

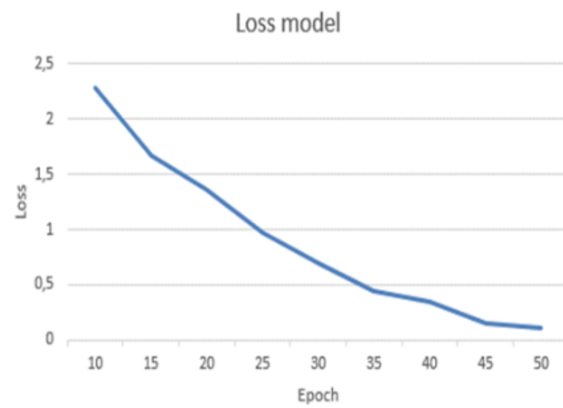

Fig. 13. Loss function on training data

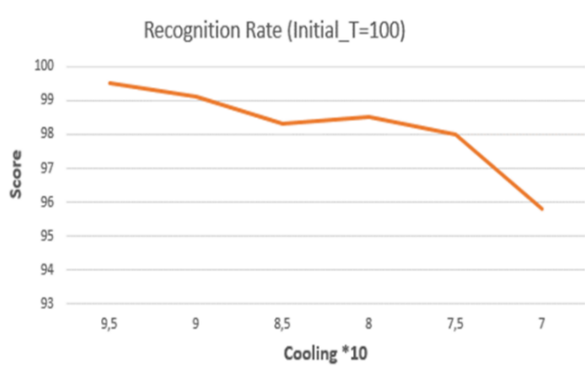

Fig. 14. Evolution of recognition rates depending on the cooling speed

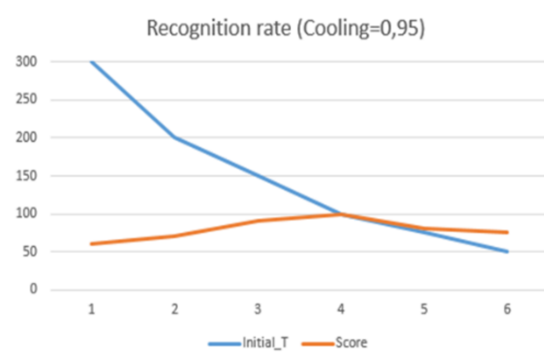

Fig. 15. Evolution of recognition rates depending on the temperature

structure is performed. Next, the learning step is based on CNN techniques and finally, a recognition step uses the SA combinatorial optimization algorithm. Our approach is, then, hybridizing: NLP + CNN + SA.

7.1 Arabic Vocabulary Structuring

In this section, we provide a detailed description of the structure we have chosen for the vocabulary. As first step, our vocabulary of decomposable words is structured as a molecular cloud, whose organization emulates the Arabic linguistic philosophy of word construction around e roots.

Each sub-cloud collects neighboring words taken from the same root based on various derivational forms, flexions and agglutinations (see Fig.6).

This factorization foregrounded common morphological entities, such as roots, schemes, with multiple conjugating elements (e.g., present, past, singular, dual, plural, masculine, feminine, etc) and agglutinations (proclitic and enclitic).

As depicted in Fig.6, the sub-cloud representing the root "**بعد**" **/baEada**/ (i.e., *to go far*) is linked to its derivatives that are presented as molecules. On the other hand, Fig.7 displays the union of different sub-clouds (black balls stand for sub-cloud nuclei). The latter correspond to the roots in order to form

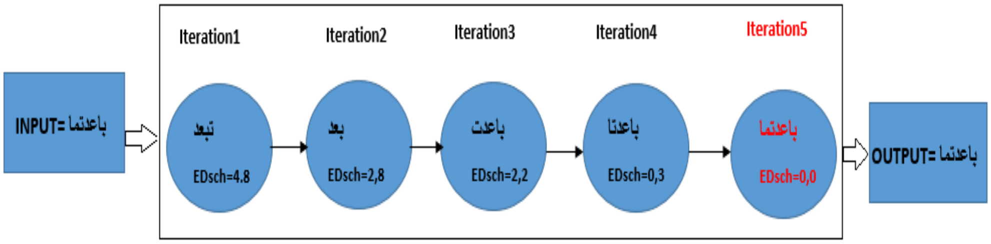

Fig. 16. A sample of successful recognition with SA algorithm

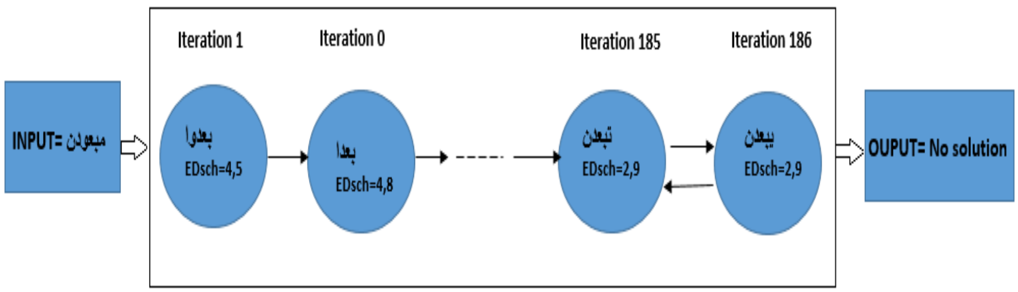

Fig. 17. A sample of misrecognition with SA algorithm

the entire molecular cloud. Thus, according to this structure, the notion of neighborhood appears two times:

On the one hand, it represents the root surrounded by all its derivatives obtained through scheme derivations forming a sub-cloud.

On the other hand, each root (a sub-cloud) is linked to a neighboring root which must be morphologically the nearest as shown in Fig.7.

This morpheme-based model of word structure enhances the learning process that characterizes most pattern recognition systems. The first step consists in preparing an initial cloud of morphologically related words by taking into account various verb conjugation rules in different tenses, with multiple persons, flexion, pronouns, and functions.

The second step focuses on incorporating some morphological instructions in order to link the generated words to sub-clouds. Finally, the first obtained cloud describes the initial solution space.

The latter is automatically stabilized in accordance with SA iterations. Note that our approach is not based on segmentation (segmentation free). It intends to integrate linguistic knowledge in both vocabulary structure and elastic comparison process.

7.2 CNN Training

As second step, based on EDsch algorithm, we tried to implement a convolutional neural network of Levenshtein distances that learns to calculate edit distances between Arabic words. Our CNN model is a set of an input and an output layer, along with numerous hidden layers, which are conventionally fully connected. The output of each layer represents the input of the next layer.

For hidden layers, they are composed of convolution layers, pooling layers followed by some fully connected layers and activation functions. The CNN model contains these different layers: (five convolution layers, two max_pooling layers, two dense layers, three dropout and one batch_normalisation layer).

In our approach, the obtained cloud is not only composed of classical nodes representing the words. It is henceforth, composed of linked neuron networks (CNN) each modeling a word. This way, we would not need an external algorithm to help searching a word into the cloud, orchestrating the whole cloud. It is indeed about giving to each CNN-node the ability to give the search while calculating itself the needed elastic distance between any word and itself. Fig.8 shows the architecture of the implemented CNN_wi that's learns to calculate the edit distance between the word wi and any other word wj for the training corpus.

7.3 Recognition by Simulated Annealing

The Simulated Annealing (SA) can be considered as a metaheuristic search algorithm with a historical overview of successful applications to numerous optimization problems over a wide range of fields [15, 16]. Based on an initial candidate solution, SA iteratively seeks for a better neighbor solution to the problem than the current one. SA and other local search algorithms (e.g., Hill Climbing) are different.

Unlike the current candidate, SA is a global algorithm that can accept a worse solution during the iterative process. Therefore, the SA method has the ability to escape from local minima and converges towards global minima in order to reach its goal [17] (see algorithm in Fig. 9). By analogy to a physical system, an optimization problem can be described based on the parameters detailed in the table 1.

The performance of simulated annealing for specific problems depends on the choice of some parameters, such as the state space, temperature, the neighbor selection method, the probability transition function and the annealing schedule [17]. These choices have a great impact on the results. These parameters will be then adjusted according to the decision rule of our SA algorithm.

Thus, SA generally provides high quality solutions. It is a general, simple and flexible method since new constraints can be easily incorporated [18]. In a previous work [19], we proposed a SA algorithm based on Levenshtein comparative method in order to recognize decomposable Arabic words.

It is a classic edit distance calculation that does not distinguish between the root consonant or scheme and the conjugated letters that occur in prefixes and suffixes. Consequently, the classical distance causes confusion when searching the solution.

To bridge these gaps, we proposed in another work [20] an edit distance calculation (EDsch), inspired by some works [21, 22], adapted to our problem by embedding linguistic knowledge into the calculation. We have incorporated new costs in the editing operations, which emphasize the presence or absence of scheme letters.

These costs differ according to the location of the scheme letters. Looking at the example presented in table 2, the classical calculation regrettably gives the same edit distance (ED = 1) face to three extremely different cases:

The first case is a comparison between the word w1="**يبعد**" /**yaboEudu**/ (derived from the scheme "**فعل**" /**faEala**/) and w2="**يبتعد**" /yabotaEidu/ (derived according to the scheme "**افتعل**" /**AfotaEala**/). Note that w1 and w2 are very different and follow two different schemes. That is why a distance equal to 1 is not satisfying.

The second case is a comparison between the word w1 and the word w3="**مبعد**" /**muboEidN**/ (derived according to the scheme "**مفعل**" /**mufuEilN**/). Once again, w1 and w3 are very distant; they are not of the same scheme class. In fact, w1 is a verb while w3 is a noun. Hence ED=1 is not acceptable.

The third case is a comparison between w1 and the word w4="**يبعده**" /**yuboEiduhu**/ (derived according to the scheme "**فعل**"). Here, indeed w1 and w4 follow the same scheme and the same flections. The ED=1 is suitable, it is in fact simply due to an agglutinated letter at the end of the same word.

Instead of that, with the new Edit distance calculation, which highlights the pattern letters, the differences between words is emphasized. In fact:

- Comparing w1="**يبعد**" and w2="**يبتعد**" gives a weighty EDsch=2,9. It is indeed due to the use of two different schemes (two different scheme letters: one at the beginning and one in the middle). The EDsch is satisfying.
- Comparing w1="يبعد" and w3="مبعد" gives a notable EDsch=2,6 (< 2,9). It is again due to the use of two different schemes (only one different scheme letter at the beginning).
- Comparing w1="يبعد" and w4="يبعده" gives a notable EDsch=1,3 (<< 2,9). The distance is fairly small since, as already mentioned, w1 and w4 follow the same scheme.

It is obvious that the idea of the algorithm focuses on the scheme letters by giving them a greater weight than the other letters (of conjugation and of agglutination). Therefore, the problem of long stagnation in local wells, that we may encounter while using Levenshtein distance, is resolved due to the integration of new costs.

This way, in this work, we propose to embed a convolutional neural network in each node of the simulated annealing, so that any node can actually calculate edit distance and participate in finding the optimal solution in the word cloud.

The proposed algorithm is an SA algorithm based on adaptive Levenshtein comparative method (EDsch) and CNN classifiers. Each node of SA is a CNN able to determine the best (EDsch) with any word. Thus, the nodes of our simulated annealing are no longer sub-clouds of words; they are sub-clouds of CNN, which are ''linguistically" trained to see the distance between Arabic words.

8 Experimentations

In order to generate our training corpus reflecting our large vocabulary, we used Arabic word images derived from tri-consonantal healthy roots, extracted from APTI (Arabic Printed Text Image) database (see Fig. 10). Which contains a large vocabulary of decomposable and non-decomposable words written in 10 fonts, 10 sizes and 4 styles (plain, italic, bold and italic combined with bold).

This combination of fonts, styles and sizes ensures a high variability of images in the database. This generated a total number of word images of over 45 million [23].

As a first step, we built our molecular word cloud corpus from the APTI word images. We organized the images in subsets according to the roots. Each subset corresponds to a root and groups all the word images derived from this one forming a molecular cloud of words. The output of this part provides various and significant information that are helpful for our work.

In addition to the Arabic linguistic features (schemes, roots, conjugating elements, etc), our corpus includes information about the neighbors of each word. For example, Fig.11 describes a neighborhood sample between the following words; "**بعد**" /**baEada**/ with Id=1 which have two direct neighbors; the word "**تبعد**" /**taboEudu**/ with Id=0 and the word "**بعدن**" /baEadona/ with Id=2 and tow indirect neighbors; the word "**تبعدن**" /**taboEudona**/ with Id=3 and the word "**أبعدن**" /**>aboEadona**/ with Id=4.

Our input corpus is a set of structural primitives extracted from Arabic word images. The process of extraction is developed in the previous work [7]. We dealt with 3148 samples derived from 11 tri-consonant roots following several schemes and extended using a variety of agglutinative and inflectional features.

8.1 CNN Results

For the CNN configurations, we choose Tensorflow and Keras frameworks running on Python. The adopted convolutional layers used 64 filters each of the size (3*3), (ReLus) as activation function, dropout technique used after each layer to minimize the overfitting, Softmax Loss and accuracy metric used for model evaluation. Padding was set to zero when needed.

The number of epochs was set to 50 in the entire architecture. To evaluate our CNN classifiers, we selected randomly around 80% of word images for training while 20% were used for test.

Table 3 shows some CNNs scores. The CNN_0/0/6/10/4/0 is relative to the word "**تبعدن**" (derived from the root "**بعد**" which Id=0, following the pattern "**فعل**" which Id=0, in the present tense which Id=6, with the second person (Id=10), etc). It can calculate the edit distance between the word "**تبعدن**" and any other word with an accuracy equal to about 98,48%.

The CNN_0/0/5/11/5/0 corresponding to the word "**بعدن**" (derived from the root "**بعد**" which Id=0, following the pattern "**فعل**" which Id=0, in the past tense which Id=5, with the third person (Id=11), etc), can predict about 99,23% of correct edit distances.

The third example is the CNN_0/2/5/10/0/0 corresponding to the word "**باعدتما**"(derived from the root "**بعد**" which Id=0, following the pattern "**فاعل**" which Id=2, in the past tense which Id=5, with the third person (Id=11), etc) can predict about

88,55%of correct edit distance. Figures 12 and 13 illustrate, respectively, the following two metrics: accuracy and loss.

8.2 Simulated Annealing Results

Experimental tests show that the highest recognition rates (99.84%) are achieved with an initial temperature equal to 100 with stable cooling equal to 0.95. Furthermore, Fig.14 and Fig.15 demonstrates that the higher the cooling rate is, the more important the recognition rate is, with initial temperature fixed at 100.

Thus, a small cooling speed allows a good recognition rate (rapid cooling returns local solutions). Figure 16 shows a scenario of successful recognition of the word "باعدتما" (went back) by simulated annealing which achieves the optimal solution after only 5 iterations.

In contrast, Fig.17 presents a scenario of misrecognition of the word "مبعودون" (are banished) by simulated annealing algorithm which stagnates after 186 iterations and does not yield an optimal solution.

9 Conclusion

In this paper, we proposed an approach that takes advantage of neural methods and combinatorial optimization techniques to recognizing a large vocabulary of decomposable Arabic words considering the particular complexity and specific nature of the Arabic script. In a first step, we structured our vocabulary by integrating Arabic linguistic knowledge in order to generate a word cloud in the form of a molecular cloud.

This structure corresponds to the Arabic concept of building words according to their roots. The sub-clouds correspond to a neighborhood of words derived from the same root. In a second step, we used CNN techniques in the learning phase with a corpus of 1500 samples of derived from one root. Experiments were carried a morphological cloud of nodes structured on sub-clouds that corresponds to tri-consonantal Arabic roots.

For corpus, we have used structural primitive vectors extracted from word images including APTI samples. Furthermore, we incorporated linguistic knowledge in the recognition phase through the SA algorithm, especially in the edit distance calculation; we have precisely highlighted the scheme letters to help the SA algorithm to achieve.

Then, each node of SA is a CNN classifier allowing to determine the best edit distance (EDsch) between two words. Thus, the nodes of our simulated annealing are no longer molecular sub-clouds of words; they are molecular sub-clouds of CNN classifiers. Our future works will focus, on increase the dataset size by integrating more roots. Then, in order to reinforce our approach, we will try to test other deep learning techniques, like LSTM and Bi-LSTM.

References

1. **Romain, R. (2019).** Contributions and perspectives on combinatorial optimization and machine learning for graph matching and classification. François Rabelais University, Tours, France.
2. **Kacem, A. K., Cheikh, I. B., Belaïd, A. (2014).** Collaborative combination of neuron-linguistic classifiers for large arabic word vocabulary recognition. International Journal of Pattern Recognition and Artificial Intelligence, Vol. 28, No. 1, pp. 1453001. DOI: 10.1142/S021800 1414530012.
3. **Cheikh, I. B., Allagui, I. (2015).** Planar Markovian approach for the recognition of a wide vocabulary of arabic decomposable words. 2015 13th International Conference on Document Analysis and Recognition ICDAR, IEEE, pp. 1031–1035. DOI: 10.1109/ ICDAR.2015.7333918.
4. **Kessentini, Y. (2009).** Modèles de Markov multi-flux pour la reconnaissance de l'écriture manuscrite multi-scripts. PhD thesis, University of Rouen.
5. **McClelland, J. L., Rumelhart, D. E. (1981).** An interactive activation model of context effects in letter perception: I. An account of basic findings. Psychological Review, Vol. 88, No. 5, pp. 375–407. DOI: 10.1037/0033-295X. 88.5.375.
6. **Belaïd, A., Choisy, C. (2008).** Human reading based strategies for off-line arabic word

recognition. In: Doermann, D., Jaeger, S. (eds) Arabic and Chinese Handwriting Recognition. SACH 2006. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg. Vol. 4768. DOI: 10.1007/978-3-540-78199-8_3.

7. **Cheikh, I. B., Zouaoui, Z. (2013).** HMM based classifier for the recognition of roots of a large canonical arabic vocabulary. In ICPRAM, pp. 244–252. DOI: 10.5220/0004335202440252.

8. **McCulloch, W. S., Pitts, W. (1943).** A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, Vol. 5, pp. 115–133. DOI: 10.1007/BF02478259.

9. **Benbakreti, S. (2019).** Développement d'un système de reconnaissance en ligne du manuscrit Arabe. Doctoral dissertation, Université de Sidi Bel Abbès-Djillali Liabes.

10. **Kanoun, S., Alimi, A. M., Lecourtier, Y. (2010).** Natural language morphology integration in off-line arabic optical text recognition. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), Vol. 41, Vo. 2, pp. 579–590. DOI: 10.1109/TSMCB.2010.2072990.

11. **Khamekhem-Jemni, S., Kessentini, Y., Kanoun, S. (2019).** Out of vocabulary word detection and recovery in Arabic handwritten text recognition. Pattern Recognition, Vol. 93, pp. 507–520. DOI: 10.1016/j.patcog.20 19.05.003

12. **Cheikhrouhou, A., Kessentini, Y., Kanoun, S. (2020).** Hybrid HMM/BLSTM system for multi-script keyword spotting in printed and handwritten documents with identification stage. Neural Computing and Applications, Vol. 32, pp. 9201–9215. DOI: 10.1007/s00521-019-04429-w.

13. **Ahmed, R., Gogate, M., Tahir, A., Dashtipour, K., Al-tamimi, B., Hawalah A., El-Affendi, M. A., Hussain, A. (2021).** Novel deep convolutional neural network-based contextual recognition of arabic handwritten scripts. Entropy, Vol. 23, No. 3, pp. 340. DOI: 10.3390/e23030340.

14. **Bello, I., Pham, H., Le, Q. V., Norouzi, M., Bengio, S. (2017).** Neural combinatorial optimization with reinforcement learning. DOI: 10.48550/arXiv.1611.09940.

15. **Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P. (1983).** Optimization by simulated annealing. Science, Vol. 220, No. 4598, pp. 671–680. DOI: 10.1126/science.220.4598.671.

16. **Huang, K. Y., Hsieh, Y. H. (2011).** Very fast simulated annealing for pattern detection and seismic. IEEE International Geoscience and Remote Sensing Symposium, pp. 499–502. DOI: doi: 10.1109/IGARSS.2011.6049174.

17. **Mahdi, W., Seyyid, A. M., Ouali, M. (2017).** Performance analysis of simulated annealing cooling schedules in the context of dense image matching. Computación y Sistemas, Vol. 21, No. 3, pp. 493–501. DOI: /10.13053/cys-21-3-2553.

18. **Xinchao, Z. (2011).** Simulated annealing algorithm with adaptive neighborhood. School of Science, Beijing University of Posts and Telecommunications, Beijing 100876, China, Applied Soft Computing, Vol. 11, No. 2, pp. 1827–1836. DOI: 10.1016/j.asoc.2010.05.029.

19. **Zeineb, Z., Imen, B. C., Mohamed, J. (2018).** Combinatorial optimization approach for arabic word recognition. Proceedings of the 2018 International Conference on Artificial Intelligence and Pattern Recognition, pp. 15–20. DOI: 10.1145/3268866.326888.

20. **Zouaoui, Z., Ben-Cheikh, I., Jemni, M. (2019).** Combinatorial optimization approach for arabic word recognition based on adaptive simulated annealing. In: Nyström, I., Hernández Heredia, Y., Milián Núñez, V. (eds) Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, CIARP 2019, Lecture Notes in Computer Science, Vol 11896. Springer, Cham. DOI: 10.1007/978-3-030-33904-3_45.

21. **Gueddah, H. (2017).** La correction orthographique des textes arabes: Contribution à la résolution d'ordonnancement et de l'insuffisance des lexiques. PhD thesis, University of Mohamed V, Rabat.

22. **Abdellah, Y., Lhoussain, A. S., Hicham, G., Mohamed, N. (2020).** Spelling correction for the arabic language space deletion errors.

Procedia Computer Science, Vol. 177, pp. 568–574. DOI: 10.1016/j.procs.2020.10.080.

23. **Slimane, F., Ingold, R., Hennebert, J. (2017).** Competition on multi-font and multi-size digitally represented arabic text. ICDAR, pp. 1466-1472. DOI: 10.1109/ICDAR.2017.239.

24. **Tu, Z., Zheng, S. Yuille, A. (2008).** Shape matching and registration by data-driven EM. Computer Vision and Image Understanding, Vol. 109, No. 3, pp. 290–304. DOI: 10.1016/j.cviu.2007.04.004.

25. **Yin, P. Y. (2000).** A tabu search approach to polygonal approximation of digital curves. International Journal of Pattern Recognition and Artificial Intelligence, Vol. 14, No. 2, pp. 243–255. DOI: 10.1142/S0218001400000167.

Article received on 26/09/2022; accepted on 05/05/2024.
** Corresponding author is Zeineb Zouaoui.*