IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 18, 2020, accepted January 12, 2021, date of publication January 22, 2021, date of current version February 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3053618

Deep Sparse Auto-Encoder Features Learning for Arabic Text Recognition

NAJOUA RAHAL[1,2], (Member, IEEE), MAROUA TOUNSI[2], (Member, IEEE), AMIR HUSSAIN[3], (Senior Member, IEEE), AND ADEL M. ALIMI[2], (Senior Member, IEEE)

[1]Faculty of Sciences of Tunis, Tunis El Manar University, Tunis 2092, Tunisia
[2]REsearch Groups in Intelligent Machines (REGIM-Laboratory), National Engineering School of Sfax (ENIS), University of Sfax, Sfax 3038, Tunisia
[3]School of Computing, Edinburgh Napier University, Edinburgh EH10 5DT, U.K.

Corresponding author: Najoua Rahal (najoua.rahal.tn@ieee.org)

This work was supported by the Ministry of Higher Education and Scientific Research of Tunisia under Grant LR11ES48.

ABSTRACT One of the most recent challenging issues of pattern recognition and artificial intelligence is Arabic text recognition. This research topic is still a pervasive and unaddressed research field, because of several factors. Complications arise due to the cursive nature of the Arabic writing, character similarities, unlimited vocabulary, use of multi-size and mixed-fonts, etc. To handle these challenges, an automatic Arabic text recognition requires building a robust system by computing discriminative features and applying a rigorous classifier together to achieve an improved performance. In this work, we introduce a new deep learning based system that recognizes Arabic text contained in images. We propose a novel hybrid network, combining a Bag-of-Feature (BoF) framework for feature extraction based on a deep Sparse Auto-Encoder (SAE), and Hidden Markov Models (HMMs), for sequence recognition. Our proposed system, termed BoF-deep SAE-HMM, is tested on four datasets, namely the printed Arabic line images Printed KHATT (P-KHATT), the benchmark printed word images Arabic Printed Text Image (APTI), the benchmark handwritten Arabic word images IFN/ENIT, and the benchmark handwritten digits images Modified National Institute of Standards and Technology (MNIST).

INDEX TERMS Arabic text recognition, feature learning, bag of features, sparse auto-encoder, hidden Markov models.

I. INTRODUCTION

In our digital world, digitized documents have become an imperious demand for storing the information into the information technology system. It tries to bring a numerical solution to the challenges posed by the widespread use of information available in paper format. The numerical solution has prompted the need for Optical Character Recognition (OCR) system to convert the input text into an ASCII format code. Recently, OCR has drawn great attention for real-world applications, including classification and retrieval [1] and information extraction [2]–[5]. The success of any applications mainly depends on the OCR high accuracy. In this research work, we are interested to the Arabic OCR (AOCR). Traditionally, the existing AOCR systems have primarily focused on constrained text recognition (i.e., fixed text length, mono-font, mono-size, lexicon-based, etc.). The development of an unconstrained AOCR either handwritten or printed is a potential source of several technical difficulties. Although a printed text is more commonly governed by well-established calligraphy rules than a handwritten text, it presents challenges that need to be addressed. These are related to the cursive nature of Arabic writing, character similarities, unlimited vocabulary, use of multi-size and mixed-fonts, etc. In addition, handwriting Arabic text recognition is challenging due to the existing variability distortions, such as twisting in strokes, inexactitude in links, and overlapping. It lights also be due to problems downward slant, intra-word compactness, and inter-word separability. This variety tends to increase exponentially when multiple writers are concerned.

This diversity complicates the choice of features to extract and the recognition algorithm. The performance (speed and accuracy) of text recognition heavily relies on robust features with a rigorous recognizer that effectively fits the variety while offering a great discriminative strength.

Usually, the input image is greatly correlated and so cannot be used for recognition as in its inherent form. The feature

The associate editor coordinating the review of this manuscript and approving it for publication was Santhosh Kumar Gopalan.

extraction step is crucial to eliminate the correlation from the original image. Given its importance, choosing appropriate features is difficult and requires a significant effort. In the recent decade, the new tendency in machine learning has been the development of techniques that automatically learn script-independent features for the Arabic text recognition. In the last decade, the learned BoF becomes a greatly competitive representation. The main breakthrough that explains the strength of the BoF is the discriminative aspect of the features, which makes it robust to rotation, scaling, and deformations.

Generally, Arabic words can be recognized in two ways: segmentation-free or segmentation-based recognition. Although the segmentation phase is the first source of error recognition, most systems avoid this step and recognize the whole word. The most successful and widely used recognizer for segmentation-free Arabic text recognition are the HMMs. Markov modeling has proved to be the most efficient classifier broadly used for Arabic text recognition. It is characterized by the avoidance of explicit segmentation of words into grapheme, or character level. Moreover, HMMs offer stochastic modeling that struggles with the variability of observation sequence lengths and nonlinear deformations, performing them appropriately for an unconstrained environment of different scripts.

As HMMs have been used in different contexts, the difference is highlighted in the enhancement of feature representation resulting in a high recognition rate. However, the results are still to be improved. A trustworthy and powerful system remains a very challenging task and the main objective of pattern recognition research. In the present work, we focus on the exploitation of the deep SAE-based BoF for feature extraction to enhance and perform the recognition task. Until recently, no researchers have used the deep SAE-based BoF and HMM techniques for text recognition.

Our contribution in this article is an outstanding system for the unconstrained Arabic text recognition. It enjoys the following strengths:

1) A new deep SAE-based method was adopted to learn the visual dictionary instead of the traditional low-level visual dictionary built in a merely bottom-up way.

2) The SAE automatically learns the optimal dictionary and simultaneously assigns optimal weights to visual words for each local descriptor.

3) Although the built dictionary is small, it does not sacrifice its discriminating power.

4) BoF-deep SAE-HMM can recognize open vocabulary, irregular, variable-size, mixed-font, high and low resolution texts.

The remainder of this article is organized as follows: Section 2 describe the background of the Arabic text recognition process and the Arabic scripts challenges. Section 3 provide a literature review covering the methods that rely on the HMMs for text recognition. In section 4, we describe the used modeling fundamentals. In section 5, our research work methodology, including the SAE-based BoF for feature extraction and HMMs for recognition is discussed. The system is empirically evaluated in section 6. Finally, the main conclusions are drawn in section 7.

II. BACKGROUND

A. ARABIC TEXT RECOGNITION PROCESS

The AOCR system consists of three main steps: preprocessing, feature extraction, and classification [76].

1) PREPROCESSING

A first preprocessing phase is applied on digitized images. It tries to correct the images imperfections and to prepare them for future high-level processes. The preprocessing phase may contain many steps such as binarization, noise reduction, skew correction, normalization, among others [77].

2) FEATURE EXTRACTION

Feature extraction for text image representation is a challenging and fascinating research field in the area of pattern recognition and image processing. It is crucial to eliminate the correlation from the original image. The OCR depends heavily on the choice of methods and types of feature extraction. It remains the main step for achieving promising results and high accuracy recognition. Still, building robust features represents a challenge in text recognition. The text recognition field computes a variety of features using either statistical [52] or structural [81] approaches. Some approaches, however, compute a combination of both statistical and structural features [78], [79]. Other approaches have used the histogram of oriented gradients as a descriptor [6], [80]. The pyramid histogram of oriented gradients has also been extracted [7]. In addition, we can mention the use of derivatives as features [8]. Several other approaches can be cited here like the principal component analysis [9], the skeleton-based features [10], and the distribution of concavity features [11].

Recently, the new tendency has moved away from handcrafted features and shifted towards machine learning that would automatically learn script independent features for text recognition, particularly the Convolutional Neural Networks (CNNs) and Deep Learning (DL). They have been adapted for feature extraction in many text recognition systems. We can cite for example: scene text recognition [12], [13], video text recognition [14], and offline handwriting text recognition [15]–[17]. However, CNN-based or DL-based approaches are still deficient. Their structures impose two main requirements: i) Huge amounts of data for training because of the large number of necessary parameters to be tuned. ii) A big number of computational resources for finding the most relevant features and achieving high performances. Over the last decade, the learned BoF has become a greatly competitive representation. It does not suffer from the previously mentioned drawbacks in CNNs and DL. Firstly, the BoF is an alternative paradigm, which is computationally efficient and that can be trained with a reasonable number of samples. Secondly, it is currently widely used in the text recognition field, as it is simple, flexible, and robust. In addition, it makes the recognition system much faster when used in a professional environment and in real-time applications such as bank account numbers, processing of bank checks, and invoice analysis. The BoF framework has been adopted in texture classification [18], object classification and retrieval [19], [20], action recognition [21], writer identification and verification [22], word spotting [23], and handwritten text recognition [24]–[26]. Most of the previously mentioned methods have proven that the BoF framework performs well in computer vision and pattern recognition branches. Nevertheless, it suffers from major drawbacks, which inhibit its efficiency:

- It adopts k-means for codebook generation, which is the step that governs the quality of image representation. This algorithm is critical insofar as it can lead to the loss of information by attributing each descriptor to the closest visual word, in particular for descriptors located at the border of several visual words.

- The clusters are placed near the most frequent descriptors. However, these descriptors are not necessarily the most discriminating. Consequently, informative descriptors are ignored.

- The low-level visual dictionary, built using k-means, repudiates the spatial layout of the local descriptors. Thus, it brings up a lack of the spatial information, thus shedding a discriminative power of class separation. Neglecting the spatial patch information, when dealing with text recognition, involves a confusion between character classes. Therefore, the shapes of the same character must be discriminant.

- With a large number of descriptors, a large codebook is necessary to obtain good results. This caused the algorithm to become slower and consume more computing time.

To overcome these problems, we tried to improve the standard BoF by optimizing the step of generating the dictionary by k-means and replacing the latter with a SAE. In this article, a new machine learning architecture of the SAE was proposed for dictionary learning. It allows benefiting from the robustness of the SAE in feature representation.

The use of the SAE for codebook learning has already proven its robustness and its ability to capture the high level content of an image. This shows its power against image clutters and occlusions. The codebook generated by the SAE has proven its effectiveness in classification and recognition in unconstrained environments [34]. When applied to text recognition, the experiments showed the strength of the SAE, proving its robustness whether for printed or handwritten text recognition. The additional reasons for using the SAE are:

- Its representation learning method that can unsupervisedly encode data into a new representation while exploiting its spatial relations.

- Sparsity aims to encode entries with only the significant characteristics of the hidden layer. It captures the salient properties of local descriptors.

- Applied to the generation of the codebook, the SAE automatically learns the optimal dictionary and simultaneously assigns optimal weights to the visual words for each local descriptor.

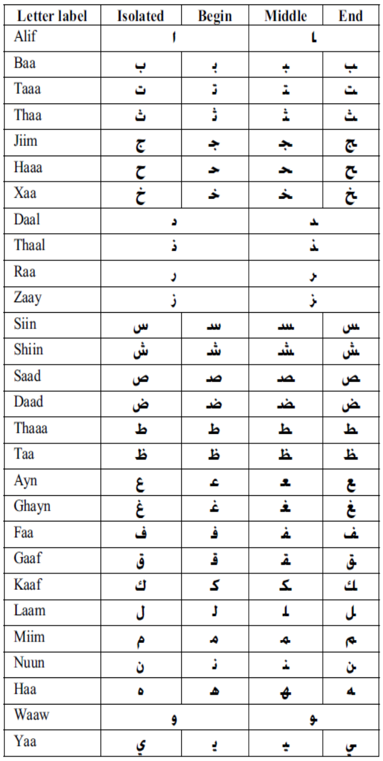
- The dictionary is built with reduced size without sacrificing its discriminating power; yet it guarantees the robustness and stability of the representations.

3) CLASSIFICATION

Classification is a required process after feature extraction i.e., the extracted features in the previous step are the input to a trained classifier. It enables us to learn a model to each character and then label it to the predicted class. There are many types of classifiers used for classifying texts in images. They are divided into two categories: a segmentation-based approach and a segmentation-free approach. The segmentation-based approach, also called the analytical approach, decomposes the text into smaller units or primary and secondary strokes (characters or pseudo-characters), in a first phase. In a second phase, it classifies the units resulting from segmentation using a classifier like multilayer perceptron (MLP) [27] or support vector machine [28] and then combines them successively providing the recognized text. However, the effectiveness of these approaches relies heavily on the results of breaking down of text into units. The segmentation-free approaches offer an alternative to beat this disadvantage. The principle of these approaches is to represent the whole text line by a sequence of feature vectors extracted from the sliding windows. HMMs [29] and Recurrent Neural Networks (RNNs) [14] are the most popular models for text recognition. HMMs and RNNs share out similarities since they comprise latent variables. However, they differ in the way of building these variables [30]. The choice of either HMMs or RNNs depends on the application domain. RNNs are computationally expensive and require large datasets because of the large number of parameters needed to be tuned for achieving a better recognition, so over-fitting may occur. Therefore, RNNs can be difficult to get working when the dataset is small. In addition, their performance deteriorates rapidly as the length of input sequence increases [31]. Contrariwise, HMMs are simpler models than RNNs. They are computationally efficient and can be trained with a little dataset. Also, they make the recognition system much faster when used in a professional environment and in real-time applications such as bank account numbers, processing of bank checks, and invoice analysis. HMMs make it possible to reconcile performance, robustness and speed. In our work, we studied and explained the theoretical background of each functionality to obtain exploitable results in a real-word data (high performance and acceptable execution time). For these reasons, using HMMs is more effective than RNNs for BoF-deep SAE-HMM system.

B. ARABIC SCRIPT CHARACTERISTICS

The main idiosyncrasies of the Arabic script may be explained as follows.



Letter label	Isolated	Begin	Middle	End
Alif	ا		ـا	
Baa	ب	بـ	ـبـ	ـب
Taa	ت	تـ	ـتـ	ـت
Thaa	ث	ثـ	ـثـ	ـث
Jiim	ج	جـ	ـجـ	ـج
Haaa	ح	حـ	ـحـ	ـح
Xaa	خ	خـ	ـخـ	ـخ
Daal	د		ـد	
Thaal	ذ		ـذ	
Raa	ر		ـر	
Zaay	ز		ـز	
Siin	س	سـ	ـسـ	ـس
Shiin	ش	شـ	ـشـ	ـش
Saad	ص	صـ	ـصـ	ـص
Daad	ض	ضـ	ـضـ	ـض
Thaaa	ط	طـ	ـطـ	ـط
Taa	ظ	ظـ	ـظـ	ـظ
Ayn	ع	عـ	ـعـ	ـع
Ghayn	غ	غـ	ـغـ	ـغ
Faa	ف	فـ	ـفـ	ـف
Gaaf	ق	قـ	ـقـ	ـق
Kaaf	ك	كـ	ـكـ	ـك
Laam	ل	لـ	ـلـ	ـل
Miim	م	مـ	ـمـ	ـم
Nuun	ن	نـ	ـنـ	ـن
Haa	ه	هـ	ـهـ	ـه
Waaw	و		ـو	
Yaa	ي	يـ	ـيـ	ـي

FIGURE 1. Arabic letters.

1) CHARACTERS USED IN ARABIC TEXT

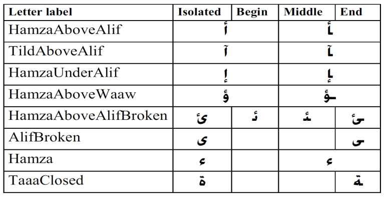
28 staple characters, 10 digits, punctuations, spaces and special symbols are used in Arabic text. Fig. 1 shows that the majority of Arabic characters undertake four different shapes according to their position in each word, i.e. Isolated (I), at the beginning (B), in the middle (M), and at the end (E). Some letters take only I and E shapes. These letters cannot be attached to the next letter; i.e., their I and B shapes are the same and their E and M shapes are the same. With all shapes, the number of letters grows and becomes 100.

Fig. 2 allows us to notice that other additional letters occurred by the variation of a few letters. Letter "Teee" is similar to letter "TaaaClosed"; however, it may be utilized right to ending an Arabic word and cannot be utilized in verbs. Additional letters are established by associating "Hamza" – "Alif" or "Hamza" – "Waaw". They are nearly pronounced similarly. Yet, their utilization relies on their positioning. Appending these letters to staple letters, the number of letters becomes 118.

Fig. 3 illustrates the variations of letter "LaamAlif" established by the association of the letter "Laam" followed by the letter "Alif" in the word. Herein, the total number of letters increases to 126.

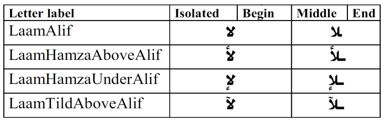
2) ARABIC LETTERS CATEGORIES

Arabic letters may be categorized into four classes: single-dot letters, two-dot letters, three-dot letters and no-dot letters, as illustrated in Fig. 4.



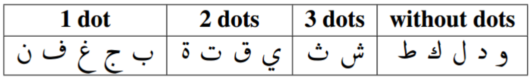
Letter label	Isolated	Begin	Middle	End
HamzaAboveAlif	أ		ـأ	
TildAboveAlif	آ		ـآ	
HamzaUnderAlif	إ		ـإ	
HamzaAboveWaaw	ؤ		ـؤ	
HamzaAboveAlifBroken	ئ	ئـ	ـئـ	ـئ
AlifBroken	ى			ـى
Hamza	ء		ء	
TaaaClosed	ة			ـة

FIGURE 2. Additional letters.



Letter label	Isolated	Begin	Middle	End
LaamAlif	لا		ـلا	
LaamHamzaAboveAlif	لأ		ـلأ	
LaamHamzaUnderAlif	لإ		ـلإ	
LaamTildAboveAlif	لآ		ـلآ	

FIGURE 3. Combined letters.



1 dot	2 dots	3 dots	without dots
ب ج غ ف ن	ي ق ت ة	ش ث	و د ل ك ط

FIGURE 4. Examples of letters with and without dot.

ب ت ث

FIGURE 5. Example of letters with selfsame shape and variable number of dots.

3) SIMILARITY OF SHAPES

Diverse Arabic letters have precisely the selfsame primordial shape. Yet, they differentiate from each other by adding dots below or beyond the baseline, as shown in Fig. 5.

4) OVERLAPPING

Adjacent letters in Arabic are mostly overlapped, which increases the difficulty of isolating letters, as illustrated in Fig. 6.

5) VARIABLE SHAPES OF THE SAME LETTER WITH DIFFERENT FONTS

Letter shapes may differ among fonts. Thus, the difficulties of the recognition task grows with the big number of existing Arabic fonts. Fig. 7 illustrates the variations of letter "Miim" in three different fonts.

6) VARIABLE SHAPES OF THE SAME LETTER WITH THE SAME FONT

In the same position and with the same font, different shapes of a letter may have. As illustrated in Fig. 8, with the font "Traditional Arabic", the letter "Baa" has five shapes at the start of the word. Its shape relies on its neighbor and the neighbor of its neighbor as in "BaaMiim" and "BaaMiimAlif".

All the aforementioned characteristics are far enough to make Arabic text recognition a complex task. Hence, more and more researchers focus on the Arabic text recognition stage. The literature has a profusion of research attempts for text recognition.

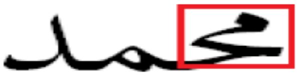


FIGURE 6. Example of adjacent overlapping letters.

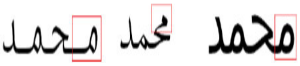


FIGURE 7. Examples of variable shapes of selfsame letter "Miim" with three fonts.

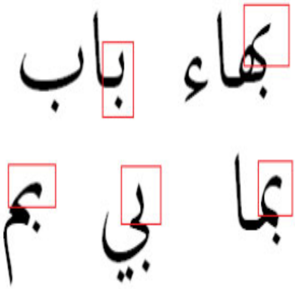


FIGURE 8. Examples of variable shapes of selfsame letter "Baa" at the start.

III. RELATED WORKS

Bazzi *et al.* [52] investigated an unlimited-vocabulary and omnifont system for Arabic and English text line recognition. For feature extraction, intensity, horizontal and vertical derivative of intensity, local slope and correlation across a window of two-cell square were all computed. The method was based on a bigram and trigram language model for unlimited-vocabulary character recognition. As results, a 3.3% Character Error Rate (CER) on the mixed-font of four fonts (Geeza, Baghdad, Kufi, and Nadim) was achieved from the DARPA Arabic OCR Corpus [70], and 1.10% on the Document English Image database from the University of Washington [71].

Reference [82] presented a system based on the Hough transform for feature extraction and HMMs for classification. Experiments were carried out using a corpus which was formed by 85000 samples in five different fonts among the most commonly used in Arabic writing, which were: Arabic Transparent, Badr, Alhada, Diwani, Koufi. 80% of samples were used as a learning base, and the rest were used for the tests. The average result achieved was 97.36%.

Natarajan *et al.* [54] presented a recognition system for Arabic, English, and Chinese scripts. Their contribution lies in the adoption of pixel percentile features which were robust to noise. A slant correction algorithm was incorporated in the preprocessing step. Features were extracted from accumulated overlapped window cells. Moreover, the angle and correlation from window cells were computed. The values were equally separated at 20 pixels perceptible and then sub joined to form a feature vector. The horizontal and vertical derivatives of these features were also attached to the feature vector. Context-dependent HMMs associated with a language model were used for recognition. They proved the efficiency of their features to recognize a text from different scripts. With the IAM English database [72], the system obtained a CER of 23.3% and a Word Error Rate (WER) of 40.1%. With the IFN/ENIT database [42], the system obtained a WER of 10.6%.

In [55], some multi-stream HMM models were used. This paradigm provided an interesting framework for the integration of multiple source of information. In the first step, preprocessing was applied to the word image. Two types of features were considered in this work: (i) contour based features and (ii) density based features. Therefore, each feature type (contour or density feature) had defined two feature streams representing the input word image. Each stream model was then separately trained using Baum Welch algorithm. The last step has been the recognition during which the HMM models were simultaneously decoded according to the multi-stream formalism. Significant experiments have been carried out on two public available databases; the recognition rate was 79.6% in IFN/ENIT for Arabic script and 89.8% in IRESTE On/Off (IRONOFF) [73] for Latin script.

Khorsheed [56] introduced a method for printed Arabic text recognition at the line level. The main contribution was the use of discrete HMMs. The pixel density features were extracted from cells falling into overlapped vertical windows. The proposed method was tested on a corpus containing 15,000 text line images written in six fonts in 600 A4-format pages. The obtained recognition rate was 95%. No peculiar case handling for mixed-font recognition was presented.

Al-Muhtaseb *et al.* [57] described an unlimited-vocabulary printed Arabic text recognition technique for eight fonts: Akhbar, Andalus, Arial, Naskh, Simplified Arabic, Tahoma,Traditional Arabic, and Thuluth. The novelty in this system was the adoption of a sliding window in a hierarchical structure. Sixteen features were extracted from vertical and horizontal overlapping and non-overlapping windows. This work was tested in the database extracted from the books of Saheh Al-Bukhari and Saheh Muslem [74], [75]. The obtained average accuracy of recognition varied between 98.08% and 99.89% for the eight fonts. The same system was used for English text recognition. The recognition result was 98.90%.

Slimane *et al.* [58] presented an open vocabulary and multi-font printed Arabic text recognition from words in low resolution. The system consisted of four main steps: font feature extraction, font recognition, word feature extraction and word recognition. For feature extraction, 102 features were extracted from each sliding window. The obtained feature vectors were then used to train the Gaussian mixture models based on the Expectation Maximization (EM) algorithm [59]. At the recognition step, an ergodic HMM model was used to allow the transitions between all the character models. The mean performances of the global multi-font system were 69.9% and 93.3% for word and character recognition, respectively. The average recognition rates were 93.7% and 98.4% using cascading (font identification). The APTI database [41], which was generated synthetically, was used for the evaluation of the developed system. However, the behavior of the system was not studied in a more challenging task, i. e. the recognition on real word images.

Reference [53] proposed a system for discovering a boundary model of small-size Arabic printed text recognition. This system was founded on state HMM number optimization and bootstrap modification to enhance accuracy by the selection of HMMs with a better performance. This approach achieved results of 86% for character recognition and 86.7% for word recognition.

Ahmad *et al.* [40] presented HMMs based on mono-font and mixed-font Arabic printed text recognition. In this work, the features were extracted from adaptive sliding windows. The text was recognized at the line level from P-KHATT database [40]. The proposed approach included two phases. In the first phase, the font of the input text line was identified, and in the second phase, the HMM was trained on the associated font for recognition. The obtained results showed the efficiency of the proposed method for mono-font. The achieved Character Recognition Rate (CRR) was between 92.45% and 98.96%. However, this method suffered from limited results with mixed-font, achieving only about 87.83%.

The previously detailed methods presented successful applications for text recognition. However, they are based on statistical and structural features, which are heavily parameterized in order to adapt to the particularities of each database. They require, for example, font and size identification systems before recognition. To cope with these problems, the Neural Networks (NNs), introduced recently, are currently the predominant technique used to learn features from images. They have effectively replaced the traditional techniques with network features and have been shown to provide significantly improved results. They are one of the fastest increasing fields in machine learning, hopeful to reshape the future of artificial intelligence. Today's state-of-the-art systems have proven that a discriminative feature extraction using NNs in combination with discriminative training of HMM models achieve excellent performance in various domains such as speech recognition [60], script identification [61], and text recognition. Many research efforts were made to design NNs architectures for feature extraction and HMMs for classification towards building a sufficient and successful text recognition system.

Rashid in [62] presented a combined Artificial Neural Network (ANN) and HMM OCR approach for degraded text line images. The proposed method adopted the MLP to extract discriminative features. Then, neural features are extracted by analyzing the contents of a contextual window. The neural features were further processed by character level HMMs to give segmentation free open vocabulary OCR. The proposed OCR system was evaluated on degraded text lines extracted from UNLV-ISRI document database [63]. A subset of 1,060 text lines was used in the evaluation step. The text lines contained major degradations like noise, broken or merged character instances and lots of variations in terms of fonts, fonts' sizes and font styles. The system achieved 98.41% CRR.

Reference [64] presented a CNN based HMMs for Arabic handwriting word recognition. In this model, the CNNs worked as a generic feature extractor and HMMs performed as a recognizer. The experiments were conducted on the IFN/ENIT dataset. The system achieved a Word Recognition Rate (WRR) of 88.95% on the training-testing "abc-d" scenario and 89.23% on the training-testing "abcd-e" scenario.

Suryani *et al.* [65] elaborated a combination of two NNs techniques, CNN and BLSTM for feature extraction followed with hybrid HMMs. The proposed system achieved a CRR of 92.00% on the HWDB 1.1 offline Chinese handwriting dataset [66].

Reference [67] proposed a Hybrid CNN-HMM system to recognize street view house numbers. The features of CNN were learnt from word level images. HMMs were used to perform training and recognition both at the whole image level without explicit segmentation. The proposed system achieved promising results of 81.07% on the SVHN dataset [68].

IV. MODELING

In this section, we detail the fundamentals of modeling that we have used. Firstly, we describe in details the principle steps of the BoF framework. Secondly, we introduce the SAE learning algorithm. Finally, we define the principle components of HMMs.

A. BoF REPRESENTATION

The BoF framework involves three main steps to convert the local descriptors to the final representation of an image: local descriptor extraction, codebook generation and quantization. In the first step, each local feature includes textures, edges, and color gradient histograms. In the second step, visual vocabulary (codebook) is generated by clustering the extracted local features, usually using K-means. Then, the obtained codebook is a set of clusters called visual word. Finally, each extracted feature for an input image is quantized to the most similar visual word in the visual vocabulary. So, every input image is reconstructed and represented as a BoF histogram. Fig. 9 illustrates the BoF paradigm. A clustering algorithm such as K-means requires calculating the distances of all feature descriptors to every visual word. Due to the large batch of features produced, it is more efficient to run the algorithms on a sparsely-sampled subset instead of complete data.

B. SPARSE AUTO-ENCODER

The SAE learning algorithm is an ANN used for unsupervised learning of discriminant features [32]. The SAE contains three layers: an Input Layer (IL) and an Output Layer (OL) connected to one (or more) Hidden Layer (HL), as illustrated in Fig. 10.

For each layer, the weight matrix is wired to the following layer. The auto-encoder reconstructs the IL via the HL.

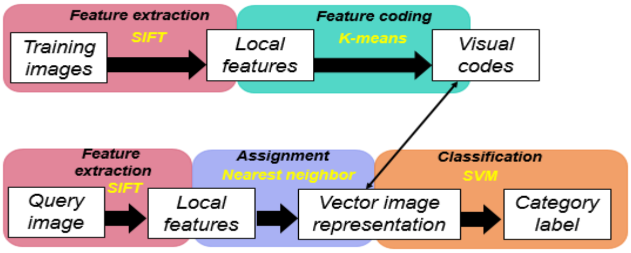


FIGURE 9. BoF Model with K-means.

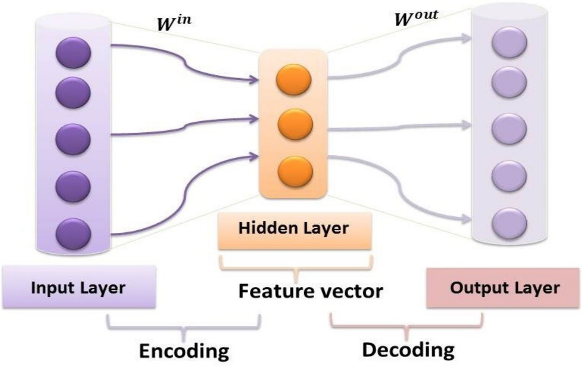


FIGURE 10. Architecture of SAE.

The sigmoid function $f(v)$ is defined as:

$$f(v) = \frac{1}{1+e^{-v}} \tag{1}$$

The decoding or the OL is given by the following equation:

$$\hat{Z}_S^k = f(\Sigma_{j=0}^{M}(\hat{W}_j \times C_j^k) + b_2) \tag{2}$$

S is the number of neurons in input and output. M is the number of neurons in the hidden layer. k is the size of the hidden layer. $\hat{W}_j$ is the weight matrix joined with linking between the HL and OL. b_2 is the bias in the OL.

C. HIDDEN MARKOV MODELS

An HMM is a statistical process with unobserved (hidden) states. The state sequence is determined through a set of observations. These are modeled state by state through the computation of emission probabilities [33].

$$P(y_t|y_1, y_2, y_3, \ldots, y_{t-1}) = P(y_t|y_{t-1}) \tag{3}$$

A chain of emissions evolves with time. For example, at time t, only O_t is observable. Yet, sequence y_t is unrecognized.

$$P(O_t|O_1 \ldots O_t, y_1 \ldots y_t) = P(O_t|y_t) \tag{4}$$

The HMM units can be presented as follows:

- $S = \{s_1, s_2 \ldots s_N\}$ is the N number of possible states. $y_t \in S$ is a state model at time t.
- $V = \{v_1, v_2 \ldots v_M\}$ is the M number of possible observations. $O_t \in V$ is an observation at time t.
- $A = \{a_{ij}\}$ is the state transition probability matrix, with:

$$a_{ij} = P(y_t = j|y_{t-1} = i), 1 \leq i, j \leq N, 0 \leq a_{ij} \leq 1 \tag{5}$$

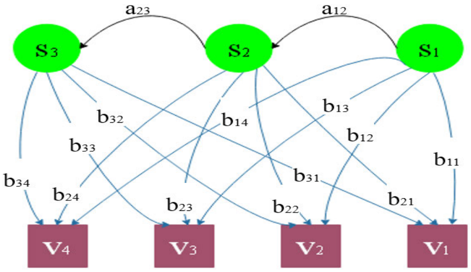


FIGURE 11. Example of HMM modeled with three states and four observations.

$$\Sigma_{j=1}^{N} a_{ij} = 1, 1 \leq i \leq N \tag{6}$$

- $B = \{b_j(k)\}$ is the observation symbol probability matrix, with:

$$b_j(k) = P(O_t = v_k|y_t = s_j), 1 \leq j \leq N, 1 \leq k \leq M \tag{7}$$

- $\pi = \{\pi_1, \pi_2 \ldots \pi_N\}$ is the initial state distribution, with:

$$\pi_i = P(y_1 = s_i), 1 \leq i \leq N \tag{8}$$

An HMM model is defined by the following triplet:

$$\lambda = (A, B, \pi) \tag{9}$$

Fig. 11 illustrates an example of an HMM modeled with three states and four observations where S represents the states, V the observations, A the transition probabilities, and B the emission probabilities.

V. DEEP SAE-BASED BoF FOR ARABIC TEXT RECOGNITION

Hereby in the present part, we detail the proposed system for Arabic text recognition. After image normalization, it starts with the Dense Scale Invariant Feature Transform (DSIFT) [35] extraction. DSIFT denotes a set of local features. The visual dictionary (codebook) is then learned using deep SAE. It is the primary process of feature coding. The quality of visual dictionary affects the effectiveness of image representation. Afterwards, for each input image, the local features are quantized using the learned codebook. Thereafter, the features are computed from an overlapped sliding windows as a BoF histograms. Histograms of the deep SAE-based BoF are presented as an input extracted features of HMMs for training and recognition. Fig. 12 describes the outline of BoF-deep SAE-HMM system and presents the listed tasks.

A. LOW-LEVEL FEATURE EXTRACTION

We start with a preprocessing step, in which the images are normalized to a fixed height, while keeping the aspect ratio. As illustrated in Fig. 12(B), DSIFT descriptors are computed over a 4 x 4 spatial grid into eight bins. Different detectors such as Harris, Harris Laplace, Hessian-Laplace,

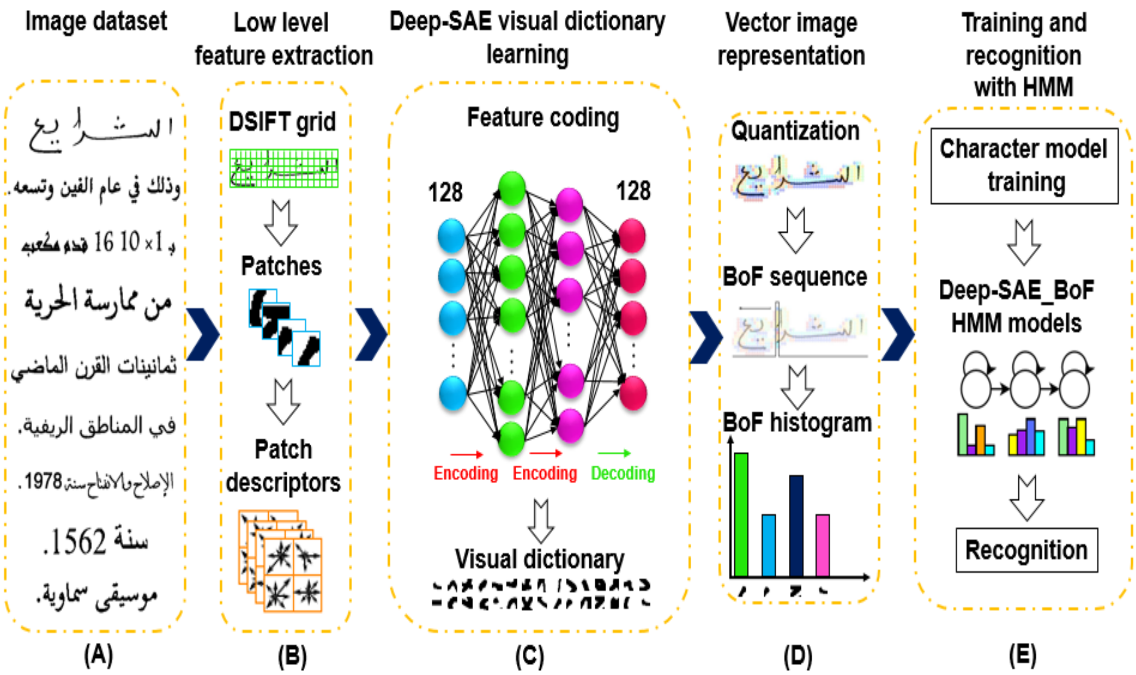


FIGURE 12. Deep SAE-based BoF-HMM Arabic text recognition system.

and Speeded up robust features have also been suggested in computer vision.

Among these detectors, the DSIFT by patch based descriptors achieved the best performance for text recognition [24].

B. SAE VISUAL DICTIONARY LEARNING

The SAE neural network is forced to learn a compressed representation of features vectors entered in inputs. As depicted in Fig. 12(C), it contains three types of layers: an IL used as an encoder, an OL used as a decoder and a HL. The IL tries to map the input feature vector $X = [x_1, x_2, \ldots, x_N]$ into the HL transformation, which is denoted as $Z = [z_1, z_2, \ldots, z_K]$, where K denotes the number of neurons used in the HL. The encoding of the input data X with size N is achieved by a non-linear activation function:

$$Z = f(WX + b) \tag{10}$$

where $W \in \Re^{K\times N}$ is a set of weights, and $b \in \Re^{K\times 1}$ is the encoding bias vector. The decoder layer function denoted as g is used to reconstruct the input from the HL. The decoding of Z is achieved using the following matrix:

$$\hat{X} = g(\hat{W}Z + \hat{b}) \tag{11}$$

where $\hat{W} \in \Re^{N\times K}$ and $\hat{b}$ represent the decoding matrix and bias, respectively.

1) SINGLE LAYER CONSTRUCTION

To optimize the SAE, the least square error as a cost function tries to find optimal parameters W,$\hat{W}$,X, which are learned by the following reconstruction error minimization:

$$\begin{aligned} E_1(W, \hat{W}; X) = \; & \frac{1}{2N}\Sigma_{i=1}^{N} \parallel \hat{x}_i - x_i \parallel_2^2 \\ & + \lambda \parallel W + \hat{W} \parallel_2^2 \\ & + \beta \Sigma_{k=1}^{K} KL(p \parallel \hat{p_k}) \end{aligned} \tag{12}$$

The SAE includes a sparsity constraint minimizing the Kullback Leibler (KL) divergence [36] on hidden layer activation. In other words, we append a regularization idiom to the reconstruction error. The KL divergence is defined by:

$$KL(p \parallel \hat{p_k}) = plog\frac{p}{\hat{p_k}} + (1-p)log\frac{1-p}{1-\hat{p_k}} \tag{13}$$

where p is the target average activation placed to the nearest centroid for data points lower than a confidence threshold, and $\hat{p_k}$ is the average activation of the hidden layer. The KL divergence represents a standard function for measuring how different two distributions are. It has the property that $KL(p \parallel \hat{p_k}) = 0$, if $\hat{p_k} = p$, otherwise it increases monotonically as $\hat{p_k}$ diverges from p. Thus, the KL divergence is used for the sparsity constraint.

2) A DEEP SAE CONSTRUCTION

We extend the single-layer architecture to be hierarchical. Therefore, we stack two SAE by stacking one additional hidden-layer in the AE with a new set of parameters $\hat{W}$, which aggregates within a neighborhood of outputs from the first AE W.

When we use the SAE for unsupervised dictionary learning, feature codes are learned from the training samples

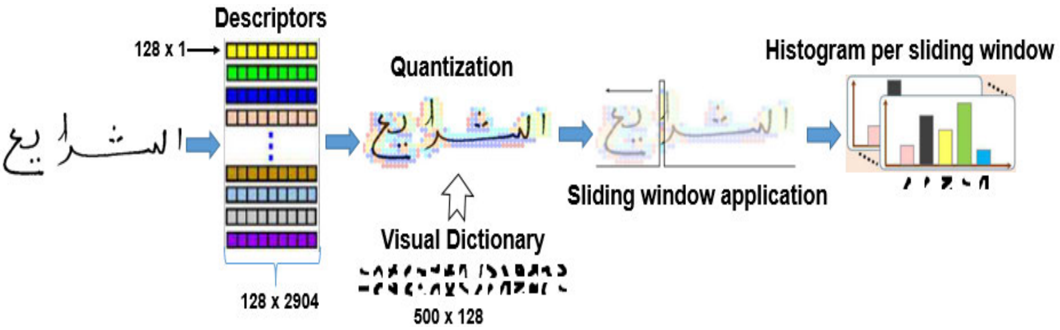


FIGURE 13. Histograms obtained by sliding window.

(the patches in our case) without considering their corresponding labels in order to explore the discriminative information existing in the patches. To further fine-tune the visual dictionary and learn the feature codes for each class used for subsequent mid-level patches, we construct a supervised two HL AE. We assign each obtained local feature code to the image label. We initialize the parameters of the features coding by values learned in the unsupervised phase and we introduce a supervision with the error back-propagation algorithm [37]. For an image dataset with L character categories, the output layer becomes a label layer with a softmax unit [38], which transforms one out of L-dimensions to L-way classification problem. Now, the mid-level visual dictionary is not only learned from the reconstructed patches, but also from a classifier predicting their labels. Let:

$$Y = y_i \in R^L; i = \{1, 2, \ldots, L\} \tag{14}$$

represents the label of L input data and $\hat{y}_i \in R^L$ is the predicted label of y_i. Our goal is to fine-tune the mid-level dictionary $W = [w_1, w_2, \ldots, w_K]$. We learn the feature codes $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_N]$ in order to minimize the average classification loss between y_i and $\hat{y}_i$ by optimizing the following objective function $E_2(W, \alpha; X, Y)$.

$$E_2(W, \alpha; X, Y) = \frac{1}{2L} \Sigma_{i=1}^{L} \parallel \hat{x}_i - x_i \parallel_2^2 \tag{15}$$

$$\hat{y}_i = softmax[\alpha(1 + exp(W^T x_i))^{-1}] \tag{16}$$

C. VECTOR IMAGE REPRESENTATION

As displayed in Fig. 12(D), the visual dictionary computed in the previous step is used to quantize the descriptors in a word image. As shown in Fig. 13, we suppose an input image having 128 x 2904 descriptors. By applying the visual dictionary with size 500 x 128, the image is represented by a histogram of 500 x 2904, which contains only the most informative features in the image. Afterwards, since HMMs are stochastic models that produce a sequence of observations, they need a sequence of BoF representations. For this purpose, a word image is divided into frames and the sequence of BoF vectors are obtained by sliding a vertical window shifted from the right to left direction. Finally, we get a set of histograms as a vector image representation.

D. TRAINING AND RECOGNITION WITH HMM

HMM modeling is implemented using the HTK [39]. It operates in two tasks: training and recognition. In both tasks, the same extracted features are used. Each image is transformed to a sequence of deep-SAE-based BoF feature histograms. For data preparation, the sequence of feature histograms is converted into a compatible format with HTK using the *HCopy* tool. The *HCompV* tool is then used to estimate the overall mean and variance vectors. The training data is used for HMM character models initialisation. We employ the linear HMM topology, where only self and next state probabilities are taken with a fixed number of states per model. An HMM is formed by the connected character sub-models of a word, as shown in Fig. 14. In the training task, the Baum-Welch iterative estimation process is implemented with the tool *HREst* to refine the Gaussian mixtures, means and variances of the parameters. At the recognition task, the extracted features from testing images are tied to HMM character models. An ergodic HMM allows the transition between character models and therefore the recognition in open vocabulary. The recognition is achieved by providing the great sequence states of characters based on the Viterbi algorithm launched by the *HVite* tool. Thus, the obtained recognition results are converted to Unicode characters. They are evaluated in terms of character and WRRs obtained by the *HResult* tool. The HTK tool *HResult* reports the performance of BoF-deep SAE-HMM system in terms of Line Recognition Rate (LRR), CRR and WER, which take into consideration the errors due to substitution, insertion and deletion. For a complete text image recognition, an ergodic HMM is used to concatenate the models. In the ergodic HMM topology, each character model may be achieved from any other character model in a finite number of transitions. Thanks to this topology, it allows the recognition in unlimited vocabulary. An example of an ergodic HMM with four character models is illustrated in Fig. 15.

VI. EXPERIMENTAL SETUP AND RESULTS ANALYSIS

A. DATASETS

To assess the performance of the extracted features, the proposed system is tested on two printed text datasets which are P-KHATT [40] and APTI [41], the handwritten Arabic text

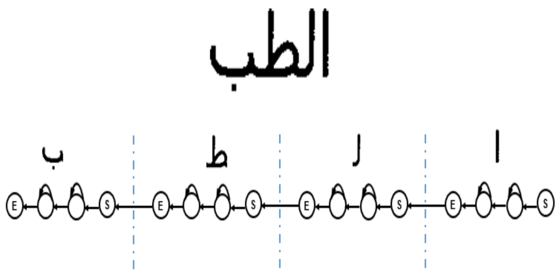


FIGURE 14. Example of HMM model for Arabic word.

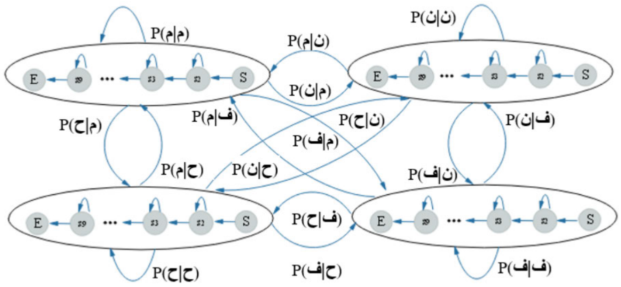


FIGURE 15. Ergodic HMM topology composed by four character models.

dataset IFN/ENIT [42], and the handwritten digits dataset MNIST [43].

1) P-KHATT DATASET

P-KHATT [40] contains text lines in eight fonts: Akhbaar, Andalus, Naskh, Simplified Arabic, Tahoma, DecoType Thuluth, Times New Roman and Traditional Arabic. The text is scanned at resolution 300 dots/inch. It includes the staple 28 Arabic letters with their different shapes and combinations, spaces, 10 digits and punctuations (".", ",", ":", ";", "!", "(", ")", "?", "-", "/", "%", etc.). It contains 6472 text line images for training, 1414 text line images for development and 1424 text line images for testing. For extra information concerning the P-KHATT database, readers can consult [40]. A few examples of text line images of this dataset are shown in Fig. 16.

2) APTI DATASET

APTI [41] consists of word images generated in 10 fonts (Arabic Transparent, Tahoma, Andalus, AdvertisingBold, Simplified Arabic, Traditional Arabic, Diwani Letter, M Unicode Sara, Naskh, and DecoType Thuluth), eight font sizes (6, 8, 10, 12, 14, 16, 18 and 24 points), and four font styles (plain, bold, italic, and combination of italic bold). It contains 113,284 words synthetically generated in low resolution with 72dpi. The ground truth is supplied in XML files. APTI is elected for the evaluation of OCR systems. Some images are shown in Fig. 17.

3) IFN/ENIT DATASET

The IFN/ENIT [42] is used in its version v2.0p1e which consists of 32,492 Arabic handwritten words having 937 Tunisian village and town names. The IFN/ENIT includes handwritten words by over 500 writers. It is divided into 5 sets a - e. We present some training - testing configurations used in our experiments: abc - d for validation and abcd - e for testing. Some images are shown in Fig. 18.

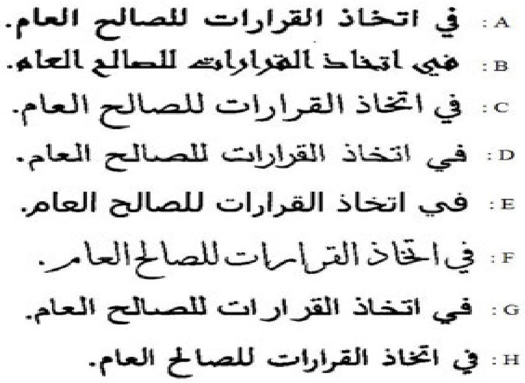

: A في اتخاذ القرارات للصالح العام.
: B في اتخاذ القرارات للصالح العام.
: C في اتخاذ القرارات للصالح العام.
: D في اتخاذ القرارات للصالح العام.
: E في اتخاذ القرارات للصالح العام.
: F في اتخاذ القرارات للصالح العام.
: G في اتخاذ القرارات للصالح العام.
: H في اتخاذ القرارات للصالح العام.

FIGURE 16. Samples of P-KHATT dataset.

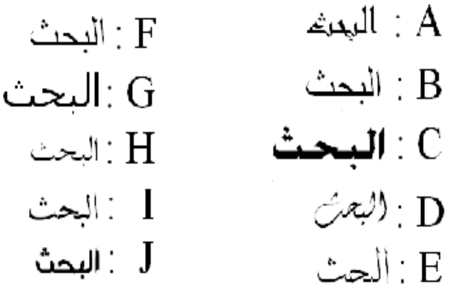

A : البحث
B : البحث
C : البحث
D : البحث
E : البحث
F : البحث
G : البحث
H : البحث
I : البحث
J : البحث

FIGURE 17. Samples of APTI dataset.

4) MNIST DATASET

MNIST [43] is a handwritten digit dataset that fits in 20 x 20 images and also normalized to fit 28 x 28 images. It was established by amalgamating the handwriting of hundreds of different writers in order to establish a greater inter-writer heterogeneity. More than 500 multiple writers contributed to the dataset, consisting of 60,000 samples for training and 10,000 for testing. A few examples of this dataset are shown in Fig. 19.

B. EXPERIMENTS

All the parameters used in P-KHATT and APTI are set based on the recognition results for the Tahoma font of the P-KHATT database.

For the IFN/ENIT database, the results are reported in two parameters (training-testing): abc-d and abcd-e. All the parameters used in IFN/ENIT and MNIST are set based on the recognition results for the IFN/ENIT dataset.

1) PATCH AND STRIDE SIZE VARIATIONS

For P-KHATT text recognition, the first step is the height normalization of 55 pixels while keeping the aspect ratio.

FIGURE 18. Samples of IFN/ENIT dataset.

FIGURE 19. Samples of MNIST dataset.

DSIFT descriptors are extracted from images in a dense patch size overlapped with a stride size.

The patch and stride sizes are very important to improve the recognition rate. They rely on image resolution. Table 1 shows three variations of the patch and stride sizes where BoF-deep SAE-HMM system reaches the best recognition rate with a patch (P) of 5 x 5 pixels and a stride (D) with 2 pixels. It is worth noting that BoF-deep SAE-HMM system provides the best accuracy recognition rate of 99.95% at the character level and 99.40% at the text line level for P=5 and D=2. Herein, it is interesting to notice that the more we increase the patch size, the more the recognition rate decreases.

For IFN/ENIT, the images are normalized to 115 pixels height while keeping the aspect ratio. Table 2 shows three variations of the patch and stride sizes where BoF-deep SAE-HMM system reaches the best recognition rate with a patch of 13 x 13 pixels and a stride of 7 pixels.

2) HL SIZE IMPACT

Different HL sizes (50, 250 and 500) are adopted to test their impact on SAE learning. In this study, we fine-tuned two main parameters: the number of HLs (1 and 2 layers) and the neurons in each HL. The experimental results indicate the great recognition rate with two HLs, 1000 neurons in the first HL and 500 neurons in the second HL. As detailed in Tables 3 and 4, we present the obtained results by adopting a 500-neuron dictionary. We get a significant WER. Beyond that point, no pertinent improvement was obtained.

TABLE 1. Patch and stride size variations (P-KHATT).

Patch size	Stride size	CRR(%)
3	1	99.88
5	**2**	**99.95**
8	4	99.69

TABLE 2. Patch and stride size variations (IFN/ENIT).

Patch size	Stride size	WER(%)
5	2	4.78
13	**7**	**1.97**
16	8	2.17

TABLE 3. Different visual dictionary sizes (P-KHATT).

HL size	CRR(%)
250	99.90
500	**99.95**
700	99.93

TABLE 4. Different visual dictionary sizes (IFN/ENIT).

HL size	WER(%)
250	3.95
500	**1.97**
700	1.96

TABLE 5. Comparison of text recognition rates via K-means and SAE.

Codebook	Tahoma		Mixed-font	
	CRR(%)	LRR(%)	CRR(%)	LRR(%)
K-means	96.70	95.18	94.70	85.33
SAE	**99.95**	**99.40**	**98.92**	**90.00**

3) SAE CODEBOOK GENERATION IMPACT

The SAE exhibits more flexibility than any hard clustering algorithm like K-means. As shown in Table 5, the obtained results demonstrate a better performance of the SAE compared to those derived via K-means. As regards the mono-font and mixed-font, the codebook generated with the SAE achieves a better performance than the codebook generated with K-means. Furthermore, for Tahoma mono-font, the best average text recognition accuracy is 99.95% for the CRR. It is improved by 3.25% compared to the k-means codebook.

4) IMPACT OF DIFFERENT NUMBERS OF GAUSSIAN MIXTURES

The basic benefit of the Gaussian mixtures is their power to model complicated shapes of probability density functions. They are modeled more precisely by increasing the number of Gaussians. Fig. 20 highlights that the CRR and LRR rates of Tahoma font text recognition are respectively increased from 97.90% and 70.35% with 1 Gaussian to 99.95% and 99.40% with 64 Gaussians. As indicated on the curve evolution, using

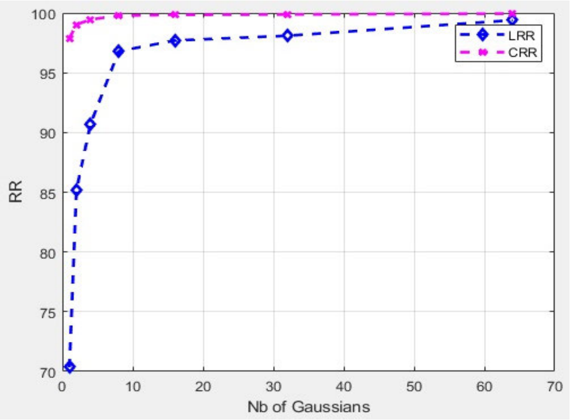


FIGURE 20. Growth of recognition rate by increasing the number of Gaussians.

TABLE 6. Imapct of number of states, W and S (P-KHATT).

W/S	Number of states	CRR(%)
3/2	10	99.87
4/1	6	99.76
4/3	**10**	**99.95**
5/2	6	99.79
6/3	6	99.77

more than 64 Gaussians does not ensure the enhancement of the obtained results.

5) IMPACT OF STATE NUMBER, SLIDING WINDOW AND OVERLAP SIZE

Tables 6 and 7 display the results of different sliding window widths (W) and overlaps (S) with several numbers of states to find the best values of character models. The optimal accuracy is obtained with parameters W=4, S=3 and the number of states=10 for P-KHATT, W=13, S=7 and the number of states=27 for IFN/ENIT.

C. COMPARISON WITH STATE-OF-ART SYSTEMS

We compare the achieved recognition results obtained by BoF-deep SAE-HMM system with those of some great related works. The reported results show that the recognition rates obtained by BoF-deep SAE-HMM system are comparable to top proposed systems.

Tables 8 and 9 show the comparison between our results and those of the state-of-the-art methods. The comparison proves the effectiveness and robustness of BoF-deep SAE-HMM system. In fact, the system proposed by [40] is validated on the P-KHATT database. The comparison is assessed using mono-font and mixed-fonts. The extracted features with the SAE show their contribution in solving the problem of morphological differences between the characteristics of characters belonging to various fonts. They allow a global and robust parameterization whatever the case of a mono-font or mixed-font context. The recorded performances are very promising in the recognition of Arabic in unconstrained environments.

TABLE 7. Impact of number of states, W and S (IFN/ENIT).

W/S	Number of states	WER(%)
3/1	35	9.32
4/3	32	5.52
5/2	30	7.27
8/4	29	3.08
8/7	**27**	**1.97**

TABLE 8. Comparison of mono-font text recognition using P-KHATT database.

Font	CER(%)	
	[40]	**BoF-deep SAE-HMM**
Times New Roman	1.20	0.10
Andalus	1.35	0.04
DecoType Thuluth	7.55	0.29
Tahoma	1.04	0.05
Traditional Arabic	4.35	0.23
Naskh	3.06	0.24
Akbaar	2.87	0.35
Simplified Arabic	1.67	0.04

TABLE 9. Comparison of mixed-font text recognition using P-KHATT database.

System	CER(%)	
	Mixed-font using samples from all fonts	Mixed-font using font identification
[40]	12.19	3.44
BoF-deep SAE-HMM	1.08	-

TABLE 10. Comparison with HMM-based text recognition using APTI database.

Feature Description	Training-Testing	CER(%)	WER(%)
DIVA-REGIM [44] Black and white connected components and their ratio, compactness, gravity centre, horizontal and vertical projection	Sets{1-5}-Set6	0.30	1.10
THOCR2 [44] Structural and statistical features and their derivatives	Sets{1-5}-Set6	0.81	4.97
[40] Statistical features and their derivatives	Sets{1;2}-Set5	0.57	2.12
BoF-deep SAE-HMM SAE-based BoF	Sets{1;2}-Set5	0.05	0.17

In Table 10, we further compare BoF-deep SAE-HMM system to the existing HMM-based systems that are reported in the literature using the APTI database. The "Arabic Transparent" is the exclusively used font in competitions that rely on such database. In fact there has been no full comparison yet for multiple reasons. Among the most important reasons is that set6, which is not publicly available, is used only to evaluate the systems in the competitions. Even the other systems that have used the APTI database and that are available in the literature, each system builds its particular training, development and evaluation sets.

TABLE 11. Evaluation of multi-size mono-font system.

System/Size		8	10	12	18	24	Mean RR
IPSAR [44]	WRR	73.3	75.0	83.1	77.1	77.5	65.3
	CRR	94.2	95.1	96.9	95.7	96.8	98.7
UPV-PRHLT-REC1 [44]	WRR	97.4	96.7	92.5	84.6	84.4	91.7
	CRR	99.6	99.4	98.7	96.9	96.0	98.3
DIVA-REGIM [44]	WRR	95.9	95.7	93.9	97.9	98.9	94.8
	CRR	99.2	99.3	98.8	99.7	99.7	99.1
BoF-deep SAE-HMM	WRR	**99.1**	**99.2**	**99.2**	**99.5**	**99.5**	**99.3**
	CRR	**99.9**	**99.9**	**99.9**	**99.9**	**99.9**	**99.9**

TABLE 12. Comparison with state-of-art systems using IFN/ENIT.

Systems	Features/Models	Training-testing configurations	
		abc-d (%)	abcd-e (%)
[29]	Local densities and statistics/HMM	21.95	-
[45]	Statistical and Structural/HMM	12.07	14.28
[46]	CNN/HMM	11.05	10.77
[47]	CNN/BLSTM	-	7.79
[26]	K-means-based BoF/HMM	3.0	5.8
Present Work	**SAE-based BoF/HMM**	**1.97**	**4.93**

Table 11 presents the system results in multi-size and mono-font Arabic word recognition. The test images assigned to the systems are the ones rendered using the font ‘‘Arabic Transparent’’, plain and sizes 8, 10, 12, 18 and 24.

Table 12 displays a comparison between the recognition results obtained by BoF-deep SAE-HMM system and others reached by well-known related work on the IFN/ENIT dataset using the same parameters (training-testing). The reported results show that the recognition rates obtained by BoF-deep SAE-HMM system are comparable to those of top proposed systems.

In what follows, we detail the most important points that prove the superiority and robustness of BoF-deep SAE-HMM system compared to the top proposed systems.

The system proposed by [48] used a writer adaptive training to train writer dependent models. Thanks to the robustness of the extracted features based on the deep SAE, BoF-deep SAE-HMM system fits effectively the variety of writers. Reference [29] achieved a WER of 21.95% for ”abc-d” despite the application of skew correction, noise reduction, size normalization and line extraction for preprocessing. In [45], the contribution was at the HMM level. The statistical and geometrical extracted features were preceded by baseline estimation. However, the proposed system obtained a WER of 12.07% for ‘‘abc-d’’ and 14.28% for ‘‘abcd-e’’. The system suffered from deformations related to the context of the character modeling. The system proposed in [49] was further developed in [46] replacing the hand-engineered feature/HMM model by the CNN/HMM model. The latter showed a WER of 11.05%, i.e. an overall decrease in the error rate with 1.02% compared to hand-engineered feature/HMMs. However, this suggested model suffered from the insufficiency of the data size used for the learning. To overcome this serious problem, the proposed system, based on CNN/BLSTM in [47], resorted to the data augmentation technique, but the obtained result, 7.79%, remained worse than the result of the system proposed by [26] using K-means-based BoF/HMMs, 5.8%. However, the low-level visual dictionary, built using K-means, repudiated the spatial layout of the local descriptors, which brought a shedding of the accuracy of separating the classes. Consequently, the results achieved by BoF-deep SAE-HMM system are better than those obtained by this system. Besides, in our case, the dictionary size was reduced to 500, reducing the computational time which dropped automatically to reach a lower value compared to [26] (dictionary size 4096).

TABLE 13. Comparison with state-of-art text recognition using MNIST database.

Systems	CER(%)
DCT-HMM [50]	2.74
BoF-Kmeans-HMM [51]	5.10
BoF-deep SAE-HMM	**1.62**

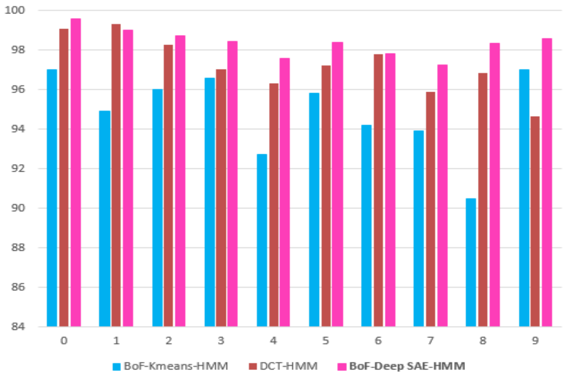


FIGURE 21. Graphical representation of comparison of BoF-deep SAE-HMM with DCT-HMM [50] and BoF-Kmeans-HMM [51].

Although the obtained results confirm the effectiveness of the SAE-based BoF HMM combination, BoF-deep SAE-HMM system made some errors in the recognition of some characters. For example, it incorrectly recognizes “ آدائهما”, in which the character ئـ” is substituted by the character “نـ”. Also, it incorrectly recognizes “آس”, where character “س” is recognized as “سن”. Anyway, the wrongly recognized characters are not very different in shape from the correct ones. In other cases, it does not detect the white space between characters e.g. “أفريقيا ،” is recognized as “ أفريقيا،”.

Digit recognition is evaluated on the MNIST dataset. The performance of BoF-deep SAE-HMM system is compared to that of the existing systems like [50] and [51]. The performance growth of BoF-deep SAE-HMM system with the state-of-the-art is presented in Table 13 and the classification graph is detailed in Fig. 21. From Table 13 and Fig. 21, it is obvious that the proposed system shows a certain stability and robustness in recognition between classes.

VII. CONCLUSION

This article contributed to the existing literature with a novel strategy for a dictionary learning phase for an efficient feature extraction using a deep SAE-based BoF. Relying on the above-mentioned method, our experimental study allowed some good results in the text recognition field. The robust features given by this system enabled us to avoid such preprocessing steps as baseline estimation and slant normalization, which are rather superfluous tasks. By greatly assigning the data to the clusters, we get dense and steady image representation. The deep SAE-based BoF learns a strong non-linear mapping in which the data are well split in the transformed space. Our analysis of the OCR literature review for Arabic script led us to select the deep SAE-based BoF as the core framework used for feature extraction and HMMs for recognition. The main goal of all our experiments was to develop a robust system able to recognize texts in an unconstrained environment. The performed experiments showed a greater robustness of the recognition system whatever the used database was.

REFERENCES

[1] N. Chen and D. Blostein, “A survey of document image classification: Problem statement, classifier architecture and performance evaluation,” *Int. J. Document Anal. Recognit.*, vol. 10, no. 1, pp. 1–16, May 2007.

[2] N. Rahal, M. Benjlaiel, and A. M. Alimi, “Incremental structural model for extracting relevant tokens of entity,” in *Proc. SMC*, 2016, pp. 3452–3456.

[3] N. Rahal, M. Benjlaiel, and A. M. Alimi, “Entity extraction and correction based on token structure model generation,” in *Proc. S+SSPR*, 2016, pp. 401–411.

[4] K. Dashtipour, M. Gogate, A. Adeel, A. Algarafi, N. Howard, and A. Hussain, “Persian named entity recognition,” in *Proc. ICCI CC*, 2017, pp. 79–83.

[5] N. Rahal, M. Tounsi, M. Benjlaiel, and A. M. Alimi, “Information extraction from Arabic and Latin scanned invoices,” in *Proc. ASAR*, 2018, pp. 145–150.

[6] A. Saidani, A. Kacem, and A. Belaid, “Arabic/Latin and machine-printed/handwritten word discrimination using HOG-based shape descriptor,” *Electron. Lett. Comput. Vis. Image Anal.*, vol. 14, no. 2, pp. 1–23, 2015.

[7] A. Saidani and A. K. Echi, “Pyramid histogram of oriented gradient for machine-printed/handwritten and Arabic/Latin word discrimination,” in *Proc. 6th Int. Conf. Soft Comput. Pattern Recognit. (SoCPaR)*, Aug. 2014, pp. 267–272.

[8] I. Ahmad and G. A. Fink, “Handwritten Arabic text recognition using multi-stage sub-core-shape HMMs,” *Int. J. Document Anal. Recognit.*, vol. 22, no. 3, pp. 329–349, Sep. 2019.

[9] K. Michal, P. Doetsch, and H. Ney, “Improvements in RWTH’s system for off-line handwriting recognition,” in *Proc. 12th Int. Conf. Document Anal. Recognit.*, Aug. 2013, pp. 935–939.

[10] G. A. Abandah, F. T. Jamour, and E. A. Qaralleh, “Recognizing handwritten Arabic words using grapheme segmentation and recurrent neural networks,” *Int. J. Document Anal. Recognit.*, vol. 17, no. 3, pp. 275–291, Sep. 2014.

[11] S. K. Jemni, Y. Kessentini, S. Kanoun, and J.-M. Ogier, “Offline Arabic handwriting recognition using BLSTMs combination,” in *Proc. 13th IAPR Int. Workshop Document Anal. Syst. (DAS)*, Apr. 2018, pp. 31–36.

[12] A. A. Chandio, M. Asikuzzaman, and M. R. Pickering, “Cursive character recognition in natural scene images using a multilevel convolutional neural network fusion,” *IEEE Access*, vol. 8, pp. 109054–109070, 2020.

[13] S. Y. Arafat and M. J. Iqbal, “Urdu-text detection and recognition in natural scene images using deep learning,” *IEEE Access*, vol. 8, pp. 96787–96803, 2020.

[14] S. Yousfi, S.-A. Berrani, and C. Garcia, “Contribution of recurrent connectionist language models in improving LSTM-based Arabic text recognition in videos,” *Pattern Recognit.*, vol. 64, pp. 245–254, Apr. 2017.

[15] S. Aly and S. Almotairi, “Deep convolutional self-organizing map network for robust handwritten digit recognition,” *IEEE Access*, vol. 8, pp. 107035–107045, 2020.

[16] R. Ptucha, F. Petroski Such, S. Pillai, F. Brockler, V. Singh, and P. Hutkowski, “Intelligent character recognition using fully convolutional neural networks,” *Pattern Recognit.*, vol. 88, pp. 604–613, Apr. 2019.

[17] R. Ahmed, K. Dashtipour, M. Gogate, A. Raza, R. Zhang, K. Huang, A. Hawalah, A. Adeel, and A. Hussain, “Offline Arabic handwriting recognition using deep machine learning: A review of recent advances,” in *Proc. BICS*, 2019, pp. 457–468.

[18] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid, “Local features and kernels for classification of texture and object categories: A comprehensive study,” *Int. J. Comput. Vis.*, vol. 73, no. 2, pp. 213–238, Jun. 2007.

[19] J. Sánchez, F. Perronnin, and T. de Campos, “Modeling the spatial layout of images beyond spatial pyramids,” *Pattern Recognit. Lett.*, vol. 33, no. 16, pp. 2216–2223, Dec. 2012.

[20] L. Zhou, Z. Zhou, and D. Hu, “Scene classification using a multi-resolution bag-of-features model,” *Pattern Recognit.*, vol. 46, no. 1, pp. 424–433, Jan. 2013.

[21] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, “Dense trajectories and motion boundary descriptors for action recognition,” *Int. J. Comput. Vis.*, vol. 103, no. 1, pp. 60–79, May 2013.

[22] S. Fiel and R. Sablatnig, “Writer identification and writer retrieval using the Fisher vector on visual vocabularies,” in *Proc. 12th Int. Conf. Document Anal. Recognit.*, Aug. 2013, pp. 545–549.

[23] L. Rothacker and G. A. Fink, “Segmentation-free query-by-string word spotting with bag-of-features HMMs,” in *Proc. 13th Int. Conf. Document Anal. Recognit. (ICDAR)*, Aug. 2015, pp. 661–665.

[24] M. O. Assayony and S. A. Mahmoud, “Recognition of Arabic handwritten words using Gabor-based bag-of-features framework,” *Int. J. Comput. Digit. Syst.*, vol. 7, no. 1, pp. 35–42, Jan. 2018.

[25] M. O. Assayony and S. A. Mahmoud, “An enhanced bag-of-features framework for Arabic handwritten sub-words and digits recognition,” *Int. J. Pattern Recognit. Artif. Intell.*, vol. 4, no. 1, pp. 27–38, 2016.

[26] L. Rothacker and G. A. Fink, “Robust output modeling in bag of features HMMs for handwriting recognition,” in *Proc. ICFHR*, 2016, pp. 199–204.

[27] A. A. Chandio, M. Pickering, and K. Shafi, “Character classification and recognition for urdu texts in natural scene images,” in *Proc. iCoMET*, 2018, pp. 1–6.

[28] N. A. Jebril, H. R. Al-Zoubi, and Q. A. Al-Haija, “Recognition of handwritten Arabic characters using histograms of oriented gradient (HOG),” *Pattern Recognit. Image Anal.*, vol. 28, no. 2, pp. 321–345, Apr. 2018.

[29] E. M. Hicham, H. Akram, and S. Khalid, “Using features of local densities, statistics and HMM toolkit (HTK) for offline Arabic handwriting text recognition,” *J. Electr. Syst. Inf. Technol.*, vol. 4, no. 3, pp. 387–396, Dec. 2017.

[30] A. Salaün, Y. Petetin, and F. Desbouvries, “Comparing the modeling powers of RNN and HMM,” in *Proc. IEEE ICMLA*, Dec. 2019, pp. 1496–1499.

[31] L. Liu, Y.-C. Lin, and J. Reid, “Comparing the performance of the LSTM and HMM language models via structural similarity,” 2019, *arXiv:1907.04670*. [Online]. Available: http://arxiv.org/abs/1907.04670

[32] A. Ng, “Sparse autoencoder,” Lect. Notes CS294A, 2011, pp. 1–19.

[33] L. R. Rabiner, ''A tutorial on hidden Markov models and selected applications in speech recognition,'' *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[34] E. Othman, Y. Bazi, N. Alajlan, H. Alhichri, and F. Melgani, ''Using convolutional features and a sparse autoencoder for land-use scene classification,'' *Int. J. Remote Sens.*, vol. 37, no. 10, pp. 2149–2167, May 2016.

[35] D. G. Lowe, ''Distinctive image features from scale-invariant keypoints,'' *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[36] S. Kullback and R. A. Leibler, ''On information and sufficiency,'' *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1951.

[37] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, ''Learning representations by back-propagating errors,'' *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[38] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*. Cambridge, MA, USA: MIT Press, 1998.

[39] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. A. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, ''The HTK book for HTK version 3.4,'' Dept. Eng., Cambridge, U.K., Cambridge Univ., 2006, p. 2011.

[40] I. Ahmad, S. A. Mahmoud, and G. A. Fink, ''Open-vocabulary recognition of machine-printed Arabic text using hidden Markov models,'' *Pattern Recognit.*, vol. 51, pp. 97–111, Mar. 2016.

[41] F. Slimane, R. Ingold, S. Kanoun, A. M. Alimi, and J. Hennebert, ''A new Arabic printed text image database and evaluation protocols,'' in *Proc. 10th Int. Conf. Document Anal. Recognit.*, 2009, pp. 946–950.

[42] M. Pechwitz, S. S. Maddouri, V. Mrgner, N. Ellouze, and H. Amiri, ''IFN/ENIT—Database of handwritten Arabic words,'' in *Proc. CIFED*, 2002, pp. 127–136.

[43] Y. LeCun, L. Jackel, L. Bottou, A. Brunot, C. Cortes, J. Denker, H. Drucker, I. Guyon, U. Muller, E. Sackinger, P. Simard, and V. Vapnik, ''Comparison of learning algorithms for handwritten digit recognition,'' in *Proc. ICANN*, 1995, pp. 53–60.

[44] F. Slimane, S. Kanoun, H. E. Abed, A. M. Alimi, R. Ingold, and J. Hennebert, ''Arabic recognition competition: Multi-font multi-size digitally represented text,'' in *Proc. 11th Int. Conf. Document Anal. Recognit.*, 2011, pp. 1449–1453.

[45] R. Mouhcine, A. Mustapha, and M. Zouhir, ''Recognition of cursive Arabic handwritten text using embedded training based on HMMs,'' *J. Electr. Syst. Inf. Technol.*, vol. 5, no. 2, pp. 245–251, Sep. 2018.

[46] M. Amrouch, M. Rabi, and A. El Mezouary, ''An efficient Arabic HMM system based on convolutional features learning,'' in *Proc. ICCSRE*, 2019, pp. 1–5.

[47] M. Rania and K. Monji, ''Convolutional neural network and BLSTM for offline Arabic handwriting recognition,'' in *Proc. ACIT*, 2018, pp. 1–6.

[48] P. Dreuw, D. Rybach, G. Heigold, and H. Ney, ''RWTH OCR: A large vocabulary optical character recognition system for Arabic scripts,'' in *Guide to OCR for Arabic Scripts*, V. Margner and H. EI Abed, Eds. London, U.K.: Springer, 2012, pp. 215–254.

[49] M. Rabi, M. Amrouch, and Z. Mahani, ''Recognition of cursive Arabic handwritten text using embedded training based on hidden Markov models,'' *Int. J. Pattern Recognit. Artif. Intell.*, vol. 32, no. 1, Jan. 2018, Art. no. 1860007.

[50] S. S. Ali and M. U. Ghani, ''Handwritten digit recognition using DCT and HMMs,'' in *Proc. ICFIT*, 2014, pp. 303–306.

[51] L. Rothacker, ''Learning bag-of-features representations for handwriting recognition,'' Diploma thesis, Dept. Comput. Sci., Technique Dortmund Univ., Dortmund, Germany, 2011.

[52] I. Bazzi, R. Schwartz, and J. Makhoul, ''An omnifont open-vocabulary OCR system for English and Arabic,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 6, pp. 495–504, Jun. 1999.

[53] Z. Jiang, X. Ding, L. Peng, and C. Liu, ''Modified bootstrap approach with state number optimization for hidden Markov model estimation in small-size printed Arabic text line recognition,'' in *Proc. MLDM*, 2014, pp. 437–441.

[54] P. Natarajan, R. S. Z. Lu, E. MacRostie, and K. Subramanian, ''Multilingual offline handwriting recognition using idden Markov models: A script-independent approach,'' in *Proc. SACH*, 2006, pp. 231–250.

[55] Y. Kessentini, T. Paquet, and A. Ben Hamadou, ''Off-line handwritten word recognition using multi-stream hidden Markov models,'' *Pattern Recognit. Lett.*, vol. 31, no. 1, pp. 60–70, Jan. 2010.

[56] M. S. Khorsheed, ''Offline recognition of omnifont Arabic text using the HMM ToolKit (HTK),'' *Pattern Recognit. Lett.*, vol. 28, no. 12, pp. 1563–1571, Sep. 2007.

[57] H. A. Al-Muhtaseb, S. A. Mahmoud, and R. S. Qahwaji, ''Recognition of off-line printed Arabic text using hidden Markov models,'' *Signal Process.*, vol. 88, no. 12, pp. 2902–2912, Dec. 2008.

[58] F. Slimane, S. Kanoun, J. Hennebert, A. M. Alimi, and R. Ingold, ''A study on font-family and font-size recognition applied to Arabic word images at ultra-low resolution,'' *Pattern Recognit. Lett.*, vol. 34, no. 2, pp. 209–218, Jan. 2013.

[59] A. P. Dempster, N. M. Laird, and D. B. Rubin, ''Maximum likelihood from incomplete data via the EM algorithm,'' *J. Roy. Stat. Soc., B, Methodol.*, vol. 39, no. 1, pp. 1–22, 1977.

[60] X. Sun, Q. Yang, S. Liu, and X. Yuan, ''Improving low-resource speech recognition based on improved NN-HMM structures,'' *IEEE Access*, vol. 8, pp. 73005–73014, 2020.

[61] A. Cheikhrouhou, Y. Kessentini, and S. Kanoun, ''Hybrid HMM/BLSTM system for multi-script keyword spotting in printed and handwritten documents with identification stage,'' *Neural Comput. Appl.*, vol. 32, no. 13, pp. 9201–9215, Jul. 2020.

[62] S. F. Rashid, ''Optical character recognition—A combined ANN/HMM approach,'' Diploma thesis, Univ. Kaiserslautern, Kaiserslautern, Germany, 2014.

[63] K. Taghva, T. A. Nartker, J. Borsack, and A. Condit, ''UNLV-ISRI document collection for research in OCR and information retrieval,'' *Proc. SPIE*, vol. 3967, pp. 157–164, Dec. 1999.

[64] M. Amrouch, R. Mouhcine, and A. El-Mezouary, ''An efficient Arabic HMM system based on convolutional features learning,'' in *Proc. ICCSRE*, 2019, pp. 1–5.

[65] D. Suryani, P. Doetsch, and H. Ney, ''On the benefits of convolutional neural network combinations in offline handwriting recognition,'' in *Proc. ICFHR*, 2016, pp. 193–198.

[66] C.-L. Liu, F. Yin, D.-H. Wang, and Q.-F. Wang, ''CASIA online and offline Chinese handwriting databases,'' in *Proc. Int. Conf. Document Anal. Recognit.*, Sep. 2011, pp. 37–41.

[67] Q. Guo, F. Wang, J. Lei, D. Tu, and G. Li, ''Convolutional feature learning and hybrid CNN-HMM for scene number recognition,'' *Neurocomputing*, vol. 184, pp. 78–90, Apr. 2016.

[68] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, ''Reading digits in natural images with unsupervised feature learning,'' in *Proc. NIPS Workshop Deep Learn. Unsupervised Feature Learn.*, 2011, pp. 1–9.

[69] R. Smith, ''An overview of the tesseract OCR engine,'' in *Proc. 9th Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2007, pp. 629–633.

[70] R. B. Davidson and R. L. Hopley, ''Arabic and Persian OCR training and test data sets,'' in *Proc. Symp. Document Image Understand. Technol.*, 1997, pp. 303–307.

[71] I. T. Phillips, S. Chen, and R. M. Haralick, ''CD-ROM document database standard,'' in *Proc. 2nd Int. Conf. Document Anal. Recognit. (ICDAR)*, 1993, pp. 478–483.

[72] U.-V. Marti and H. Bunke, ''The IAM-database: An English sentence database for offline handwriting recognition,'' *Int. J. Document Anal. Recognit.*, vol. 5, no. 1, pp. 39–46, Nov. 2002.

[73] C. Viard-Gaudin, P. M. Lallican, S. Knerr, and P. Binter, ''The IRESTE on/off (IRONOFF) dual handwriting database,'' in *Proc. 5th Int. Conf. Document Anal. Recognit. (ICDAR)*, 1999, pp. 455–458.

[74] M. Al-Bukhari, ''Al-Jame' Al-Saheeh (Sahih Al-Bukhari),'' Dar Al-Jeel, Beirut, Lebanon, Tech. Rep., 2005.

[75] M. Al-Naysabouri, ''Al-Jame' Al-Saheeh (Sahih Muslim),'' Dar Al-Jeel, Beirut, Lebanon, Tech. Rep., 2006.

[76] A. Rehman and T. Saba, ''Neural networks for document image preprocessing: State of the art,'' *Artif. Intell. Rev.*, vol. 42, no. 2, pp. 253–273, Aug. 2014.

[77] A. M. Namboodiri and A. K. Jain, ''Document structure and layout analysis,'' in *Digital Document Processing*. London, U.K.: Springer, 2007, pp. 29–48.

[78] E. M. Hicham, H. Akram, and S. Khalid, ''Using features of local densities, statistics and HMM toolkit (HTK) for offline Arabic handwriting text recognition,'' *J. Electr. Syst. Inf. Technol.*, vol. 4, no. 3, pp. 387–396, Dec. 2017.

[79] A. T. Sahlol, M. A. Elaziz, M. A. A. Al-Qaness, and S. Kim, ''Handwritten Arabic optical character recognition approach based on hybrid whale optimization algorithm with neighborhood rough set,'' *IEEE Access*, vol. 8, pp. 23011–23021, 2020.

[80] A. Al-Dmour and M. Abuhelaleh, ''Arabic handwritten word category classification using bag of features,'' *J. Theor. Appl. Inf. Technol.*, vol. 89, no. 2, p. 320, 2016.

[81] R. A.-H. Mohamad, L. Likforman-Sulem, and C. Mokbel, "Combining slanted-frame classifiers for improved HMM-based Arabic handwriting recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 7, pp. 1165–1177, Jul. 2009.

[82] N. B. Amor and N. E. B. Amara, "Multifont Arabic characters recognition using HoughTransform and HMM/ANN classification," *J. Multimedia*, vol. 1, no. 2, pp. 50–54, May 2006.

NAJOUA RAHAL (Member, IEEE) is currently pursuing the Ph.D. degree with Tunis El Manar University, Tunisia. She is currently a member of the REsearch Group in Intelligent Machines (REGIM-Laboratory). Her research interests include document analysis, pattern recognition, text recognition, and information extraction.

MAROUA TOUNSI (Member, IEEE) received the Ph.D. degree in 2020. She is currently a member of the REsearch Group in Intelligent Machines (REGIM-Laboratory). Her research interests include document analysis, image processing, pattern recognition, and character recognition.

AMIR HUSSAIN (Senior Member, IEEE) received the B.Eng. (Hons.) and Ph.D. degrees from the University of Strathclyde, Glasgow, U.K., in 1992 and 1997, respectively. Following postdoctoral and academic positions at the Universities of West of Scotland (an EPSRC Postdoctoral Fellow from 1996 to 1998), (a Research Lecturer from 1998 to 2000), Dundee, and (a Lecturer from 2000 to 2004; a Senior Lecturer from 2004 to 2008; a Reader from 2008 to 2012; and a Professor from 2012 to 2018), Stirling. He joined Edinburgh Napier University, Scotland, U.K., in 2018, as a Professor and the Founding Head of the Data Science and Cyber Analytics (DSCA) Research Group (more than 20 academics and research staff). He is also the founding Head of the Cognitive Big Data and Cybersecurity (CogBiD) Research Laboratory.

ADEL M. ALIMI (Senior Member, IEEE) graduated in electrical engineering in 1990. He received the Ph.D. and HDR degrees in electrical and computer engineering in 1995 and 2000, respectively. He has been a Full Professor in electrical engineering with the National Engineering School of Sfax (ENIS), University of Sfax, since 2006. He is currently the Founder and the Director of the Research REGIM-Laboratory in intelligent machines. He is also the Director of the Tunisia Erasmus+ Office, since 2018. His research interests include applications of intelligent methods (neural networks, fuzzy logic, and evolutionary algorithms) to pattern recognition, robotic systems, vision systems, and industrial processes.