Off-line Arabic Handwriting Recognition Using Dynamic Random Forests

Kawther Rouini, Khawla Jayech, Mohamed Ali Mahjoub
LATIS – Laboratory of Advanced Technology and intelligent Systems ENISO
University of Sousse
Tunisia

Abstract

Arabic handwriting recognition is still a challenging task related especially to the unlimited variation in human handwriting, to the presence of overlapping and ligature between characters and to the high variety of Arabic character shapes. In this paper, we present an offline Arabic handwriting recognition system based on Random Forests (RFs). In fact, the RFs are a successful technique for classification and have a lot of advantages compared to other classifiers proposed in the literature to wit (i) the very high classification and recognition accuracy, (ii) the ability to determine the variable importance, and (iii) the flexibility to perform several types of statistical data analysis, including regression, classification, survival analysis, and unsupervised learning. In this study, we put forward an approach using the Surf descriptor to extract reliable features and a new RF induction algorithm called the Dynamic RFs (DRFs), which has the advantage of efficiently modeling the interaction among trees to determine the right prediction. The DRF is based on an adaptative tree induction procedure. The results carried out on the Tunisian city names database show that the DRF proves a significant improvement in terms of accuracy compared to the standard static RF and it reduces the computational time as well.

***Keywords**: Dynamic Random Forests, Arabic handwriting recognition, Surf descriptor.*

1. Introduction

Arabic handwriting recognition has been one of the main interests of computer vision researchers [1, 2]. A lot of work has been done, but this task is still an active and challenging problem due to the high variability and complexity of the Arabic script, which is manifested by (i) the unlimited variation in human handwriting, (ii) the similarities of character shapes and their position in the word, (iii) the ability of diacritical marks to be fused or not allocated exactly under or above the letter body, and (vi) the possibility for the Arabic words to horizontally overlap and for letters to stack on others [1, 3, 4, 5, 6]. In an attempt to solve this problem, different models have been proposed especially based on the classifiers of probabilistic graphical models, like the hidden Markov models and the Bayesian Networks [1,2], and on Neural Networks (NNs) such as the Recurrent NNs and the Dynamic NNs [7]. Recently, in the pattern recognition field, growing interest has been shown for multiple classifier systems, mainly for boosting, bagging and random-subspace classifiers. Those methods aim to induce an ensemble of classifiers by producing diversity at different levels [8]. Following this principle, Breiman in [9] proposed another family of methods, called the Random Forests (RFs). The RF classifiers are an effective and popular ensemble of classifiers derived from the decision tree idea, which uses majority voting or averaging as a combination function and applies them in various classification and regression domains [10]. Actually, the RF interests are (1) a novel technique using to identify the variable importance, (2) a faculty to model complex interactions among predictor variables, and (3) an approach for imputing missing data. Nevertheless, it is clear that each tree in an RF may have a different contribution in processing a certain instance. Several RF extensions have been suggested in the literature in order to improve the combination of trees by taking into account their local performance. In this paper, we propose a new RF extension called the Dynamic RFs (DRFs) so as to increase the performance of the offline Arabic handwriting recognition system.

The rest of the paper is organized as follows. In section 2, we present a brief state-of-the-art of the related work. Then, section 3 reviews the principle of RFs and details the suggested new DRFs. In section 4, we develop the architecture of the proposed system. In section 5, we present and discuss the results of our experiments. Finally, the conclusions and suggestions for future work directions are drawn in the last section.

2. Related work

The review of the literature shows limited research work, which uses the RFs for handwriting recognition.

However, those classifiers show good results in the different domains of pattern recognition.

Abdeen in [11] put forward an approach for Arabic handwriting word segmentation and recognition based on the RFs. In fact, after the segmentation process, a set of zoning features were extracted and used to train efficient RF classifiers. The extracted features had the advantage of efficiently modeling the local and global characteristics of handwritten characters. The proposed approach is tested using the IFN/ENIT database and is compared to NNs classifier. The results prove the accuracy of the RFs and their ability to outperform the performance of the NNs.

Bernard in [8] developed an approach for character handwritten recognition using the RFs. The aim of their work was to study those methods in a strictly pragmatic approach, in order to provide rules on parameter settings for practitioners. For that purpose, they were experimented the Forest-RI algorithm, considered as the RF reference method, on the MNIST handwritten digits database. The results demonstrated the efficiently of such a RF model for character handwriting recognition.

Rashad in [12] suggested an investigation of using both K- Nearest Neighbor (KNN) and RF Tree (RFT) classifiers with previously tested statistical features. These features were independent from the fonts and size of the characters. First, a binarization procedure was performed on the input characters images, and then the main features were extracted. The used features were statistical ones calculated on the shapes of characters. A comparison between the KNN and RFT classifiers was evaluated. The RFT was found to be better than the KNN by a recognition rate more than 11 %. The effect of the different parameters of these classifiers was also tested, as well as the effect of noisy characters.

Zamani in [13] developed a handwritten digit recognition system based on the RFs and the Convolutional NNs (CNNs). They performed some experiments with different preprocessing steps, feature types, and baselines. The results proved that the RFs performed better than the CNNs.

Ahmed et al. [18] employed a special type of recurrent NN, called Bidirectional Long Short-Term Memory (BLSTM) networks, to propose a segmentation-free optical character recognition system. The BLSTM evinced its efficiency in a lot of research areas thanks to its ability to remember events when there were long time lags between events. However, the BLSTM required pre-segmented training data and post-processing to transform the outputs into label sequences. Therefore, a layer, called the connectionist temporal classification was used with the BLSTM to label unsegmented sequences directly. The proposed approach was evaluated with cursive Urdu and non-cursive English scripts. However, their experiments showed that the accuracy of the proposed approach was 11.99% with non-cursive scripts and 99.18% with cursive ones. Hence, the suggested approach needed more investigation to enhance its accuracy with cursive scripts.

This brief state-of-the-art showed the efficiency of using the RFs in an OCR compared with other classifiers such as the NN and the KNN.

3. Dynamic Random Forests

3.1. Random Forests

An RF is a very successful and popular algorithm used in classification domains. RF is based on the combination of several trees, and its output F(n) is the average of the output of all trees. An RF is an ensemble of decision trees trained with random features. In this case, each tree is trained by adding new stumps in the leaf nodes where a maximum information gain can be achieved. The construction of the Tree is made by algorithm 1 and the formal definition and the use of the term RF was introduced by Leo Breiman in [9].

Rnd-Tree

Input
n : current node,
T : data set associate d to node n,
K: number of randomly selected variables on each node.
Output: n
1: **if** n is not a leaf
2: C ← K randomly selected variables
3: **for** A ∈ C **do**
4: Procedure CART
5: Partition ← Partition which optimize the Gini criterion
6: n.addchild(partition)
7: **for** child ∈ n.childnode **do**
8: RndTree(child, child.data, K)
9: **return** n

Algorithm 1: Tree construction steps

In algorithm 2, we introduce the procedure to construct the RFs

Forest-RI

Input
T: dataset for training
L: number of trees
K: number of randomly selected variables on each node.
Output: RF
Begin
1: **for** l=1 : L **do**
2: T_l ← bootstrap set
3: tree ⟵ empty tree
4: tree. root⟵Rndtree (tree.root, T_l , K)
5: RF ⟵ RF ⋃ tree
6: return RF
End

Algorithm 2: Forest-RI

Complexity of the Forest_RI algorithm:

$$C(\text{forest\_RI}) = \Theta(L) \times C(\text{RndTree})$$
$$C(\text{forest\_RI}) = \Theta((L \times N \times k \times \log(N))$$

L is the number of trees, N is the number of data, and K is the number of randomly selected characteristics.
In order to improve the performance of the RFs, many extensions have been proposed in the literature, denoted by the DRFs and defined in the next section.

3.2. Dynamic Random Forest

Simon Bernard defined a new RF, named the DRF and based on adaptative weighting induction of trees. The main idea of this new algorithm is to adapt the induction of each tree to the current forest to construct a different set whose members cooperate well with others.

The DRF uses a sequential and dynamic procedure to construct a set of random trees. The main idea for Bernard is to force the induction of the next tree to focus on the worst predicted training samples by the forest using a training data weighting. In fact, the samples that are so hard to class will be favored in the induction of the next tree, thus affecting those having an important weight. To calculate this weight, Simon Bernard used a measure of reliability of the forest's prediction which would lean on the number of votes attributed to the right class of data.

DRF

Input
A: training set,
M: number of randomly selected variables,
L: number of trees,
$\mathbf{W_\alpha(c(x,y))}$: weighting function.
Output: Random Forest
Begin
1: **for** $\mathbf{x_i} \in$ **A do**
2:$\mathbf{D_1(x_i) = 1/N}$ /// weighting vector
3: **for (l<=L) do**
4: tree ⟵ empty tree
5: Z ⟵0
6: $\mathbf{A_l}$⟵ weighted bootstrap set
7: tree.root ⟵ RndTree (tree.root, $\boldsymbol{A_l}$)
8: RandomForest ⟵ RandomForest ∪ tree
9: **for** $\boldsymbol{x_i} \in$ A **do**
10: $\boldsymbol{D_{l+1}(x_i)} \longleftarrow \mathbf{W_\alpha(c(x_i,y_i))}$
11: Z⟵ Z + $\boldsymbol{D_{l+1}(x_i)}$
12: **for** $\mathbf{x_i} \in$ **A do**
13:$\mathbf{D_{l+1}(x_i)} \leftarrow \frac{\mathbf{D_{l+1}(x_i)}}{\mathbf{Z}}$ /// normalizing weights
14: **return** Random Forest
End

Algorithm 3: DRF

Complexity of DRF:

$$C(\text{DRF}) = \Theta(L \times (2N + C(\text{Rtree})))$$
$$C(\text{DRF}) = \Theta\left(L \times \left(2N + \left(N \times k \times log(N)\right)\right)\right)$$

L is the number of trees, N is the number of data, and K is the number of randomly selected characteristics. Training with the DRF is different from training with the Forest_RI on the process of partition-rule selection. Each rule divides the group of training data into two groups based on certain evaluation criteria. These criteria in using the Forest_RI are based on the effectives of each class of problems in each group. In using the DRF, those effectives are replaced by the sum of weights of data contained in the sub groups.

4. Proposed system

4.1. System description

The architecture of the proposed recognition system is shown in figure 1, which includes two mains stages: (i) feature extraction and quantification step, and (ii) modeling and training. The features are extracted using the Surf descriptor and the Bag of Words (**BoW**) model. The RF is used afterwards to recognize an unknown word to which class is belonged.
The Speeded Up Robust Features (SURF) descriptor is a detector as well as a descriptor of key points, defined by Willems in 2008. It represents the most currently used method. The SURF is not very sensible to geometric transformations, which makes it a very robust descriptor. This algorithm is composed of four steps. The first step is the key point detection (localization) where the SURF descriptor uses the integral image and the square filters because filtering an image with square filters is the most rapid in utilizing the integral image. To detect the key point SURF uses a blob descriptor based on the hessian matrix. The second step is the space-scale representation which is realized in the form of a pyramid of images. The third step is the affectation of orientations. To find the orientation of the key point and in order to be invariant to rotation, the Haar-wavelet responses must be calculated according to the x and y direction within a circular region around the key point. Finally, the fourth step is to extract from this region the SURF description.

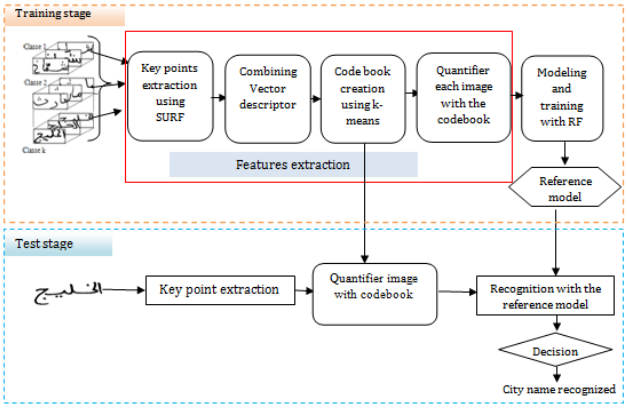

Figure 1: Architecture of proposed system

4.2. Feature extraction

Feature extraction is to extract the most essential information from images and represent the word image by a set of numerical characteristics. In this stage, after the preprocessing stage, we use the Surf descriptor surf which is a detector as well as a descriptor of key points where each point is represented by a vector of 64*1 dimensions. To efficiently manage the key points, we choose to use the BoW model. The main idea of this model is to quantifier each key point extracted in one of the visual words and then to represent each image by a histogram of visual words. The algorithm of the BoW model is based on four principal steps as shown in figure 2. Firstly, a certain number of key points are detected from each image by using the surf descriptor. Then, we regroup the descriptors' vectors of all key points extracted from all images existing in the training data. Secondly, we create the codebook by using the K-means algorithm where each column in the codebook represents a visual word. The visual word has the same dimension of that of the descriptor vector of each key point. Third, we count for each visual word the number of occurrences in the image. Finally, we represent each image by a vector of K visual words.

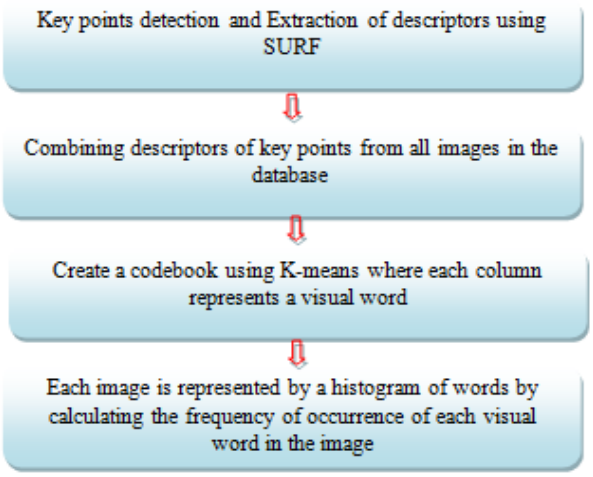

Figure 2: Feature extraction and quantification steps

4.3. Modeling and training

The stage "modeling and training" consist in classifying an unknown word and recognizing which class it belongs to. After, we extract the features from the image, as indicated in the precedent section; we define the different models used for the RF model. In this stage, we use two RF models, the static model and the dynamic one. Hence, we use the forest-RI (**forest-Random Input**) which is the first static algorithm developed by Breiman in [9]. It represents the referent algorithm for the majority of research work. We use also the DRF algorithm which represents a new dynamic algorithm defined by Bernard in [14]. Bernard, by studying the evaluation of the DRFs, showed that with a certain number of trees, we could reach a maximum performance gain. Then, adding more trees to the forest would not allow obtaining better performances in generalization. These results highlight the possibility of setting up a stop criterion, thus defining one. In fact, the main contribution of our algorithm, presented in algorithm 4 is to propose a criterion in order to stop the processes when the recognition rate does not increase in a marge of 50 trees. In addition, by studying the out-of-bag measures of the overall bagging method, we find that almost 40% of the data are not known by the tree. Subsequently, we decide to reduce the number of randomly selected (with replacement) samples to use for growing each tree to 50% where we decide to select the samples which have higher weight instead of selecting them randomly.

New DRF

Input
A: training set,
M: number of randomly selected variables,
L: number of trees,
$\mathbf{W_\alpha(c(x,y))}$: weighting function.

Output: RF
Begin
1: **for** $\mathbf{x_i} \in$ **A do**
2:$\mathbf{D_1(x_i) = 1/N}$ /// weighting vector
3 : t=false
4 : l=1
5: **while (t=false) & (l<=L) do**
6 : tree ⟵ empty tree
7: Z ⟵0
8: $\mathbf{A_l}$⟵ weighted bootstrap set
9: tree.root ⟵ RndTree (tree.root, $\boldsymbol{A_l}$)
10: RandomForest ⟵ RandomForest ∪ tree
11: Models{l}⟵ RandomForest
12: **for** $\boldsymbol{x_i} \in$ A **do**
13: $\boldsymbol{D_{l+1}(x_i)} \longleftarrow \mathbf{W_\alpha(c(x_i,y_i))}$
14 : Z⟵ Z + $\boldsymbol{D_{l+1}(x_i)}$
15 : **for** $\mathbf{x_i} \in$ **A do**
16 :$\mathbf{D_{l+1}(x_i)} \leftarrow \frac{\mathbf{D_{l+1}(x_i)}}{\mathbf{Z}}$ /// normalizing weights
17 : if (l>50)
18: if (rate(Models{l},S)-rate(Models{l-50},S)) <0.001
19: t=true
20: l=l+1
21: **return** RF
End

Algorithm 4: The developed algorithm: New DRFs

The function rate(x,y) is a function that returns the recognition rate obtained by the model x and the testing set y.

Complexity of new DRF:

$$C(\text{new}-\text{DRF}) = \Theta(L \times (N + C(\text{RndTree})))$$

$$C(\text{new}-\text{DRF}) = \Theta\left(L \times \left(2N + \left(\frac{N}{2} \times k \times log\left(\frac{N}{2}\right)\right)\right)\right)$$

5. Experimentation and results

5.1. Database description and feature extraction

In order to evaluate the performance of the proposed system, experiments are conducted on the IFN/ENIT database, which contains 946 handwritten Tunisian town/village names and their corresponding postcodes. The database consists of 32,492 samples in a new version v2.0p1e written by more than 1000 writers and divided into seven distinct sets: a, b, c, d, e, f, and s. All handwritten forms are scanned with 300dpi and converted to binary images. Afterwards, we extract the features using the Surf descriptor. Figure 3 represents the different steps which we follow to extract the features. We justify the choice of the value "4,096" for the parameter 'k' in the next section.

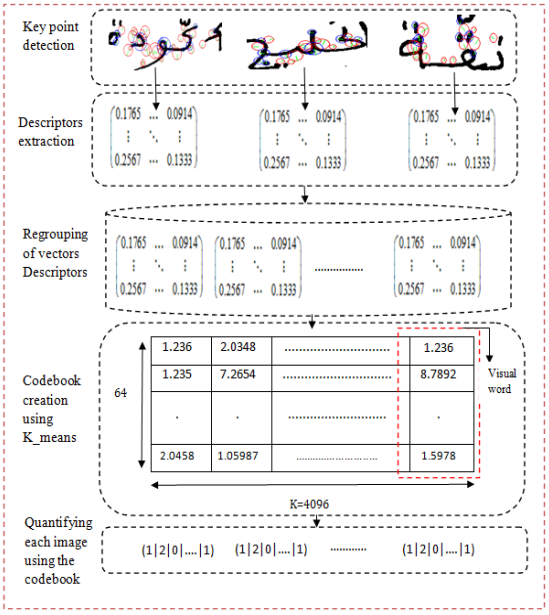

Figure 3: Feature extraction

5.2. Experimental protocol

To test the effectiveness of the new DRF model, we try to vary the input arguments which are the number of trees, the number of randomly selected variables to consider at each node and the weighting function in order to optimize our model. We represent the results of experiments in the table 1, figure 4 and figure 5.

Simon Bernard proposed three types of weighting functions (polynomial, exponential and inverse). Each function has as an input the parameter α. This parameter allows accentuating or weakening the differences in weighting between two data, which have two different values of C(x, y).

$$W_\alpha(c(x,y)) = 1 - c(x,y)^\alpha$$

$$W_\alpha(c(x,y)) = \exp(-\alpha \times c(x,y))$$

$$W_\alpha(c(x,y)) = \frac{1}{c(x,y)^\alpha}$$

We provide in table 1 the recognition rate obtained with each function and each value of α so as to optimize our model and to find the function and the value that return the best recognition rate. As it can be noted, the best results are obtained when using the polynomial function; and there will no difference in the recognition rate when we change the value of α, accordingly, we use the value 1.

f(x)	polynomial %			exponential %			inverse %		
α	1	2	5	2	10	50	1	2	5
a	90.06%	90.0%	89.65%	89.3%	89.67%	83.96%	71.33%	76.67%	81.67%
b	89.78%	89.97%	89.26%	90.06%	89.45%	88.33%	76.02%	79.67%	80.15%
c	91.03%	90.67%	88.97%	90.0%	90.06%	89.08%	79.78%	81.0%	79.01%
d	90.64%	90.3%	90.47%	89.67%	90.3%	88.0%	69.93%	77.3%	76.0%

Table 1: Recognition rate obtained with different function

As it is illustrated in figure 4 and 5, the recognition rate increases with the number of trees to obtain a sort of stability and the time execution grows with the size of bootstrap samples. We try with those experiments to find a compromise between the recognition rate and the time execution. We find that the best performance obtained with 800 trees and 13 variables.

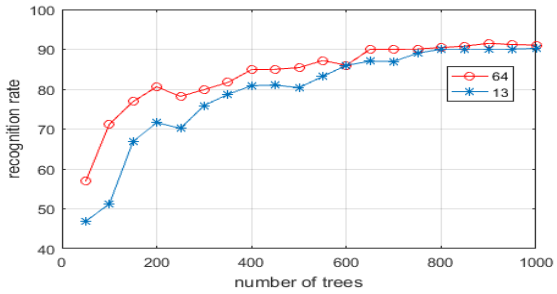

Figure 4: Recognition rate vs number of trees and number of randomly selected characteristics

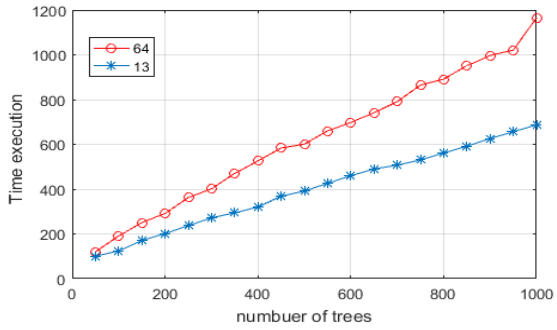

Figure 5: Execution time vs. number of trees and number of randomly selected variables

Figure 6 shows that the recognition rate goes up with the codebook size, and we find that the best rate is obtained with a codebook size of 4096.

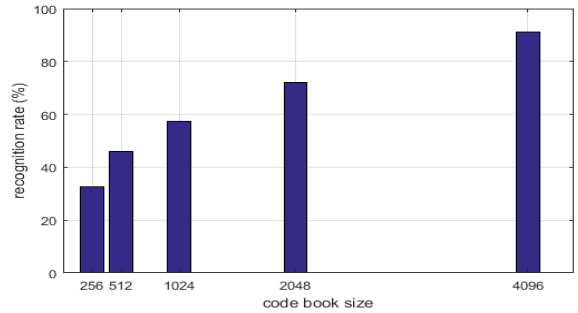

Figure 6: Recognition rate vs. codebook size

Table 2 indicates that the DRF returns a better recognition rate than the static RF. A dynamic induction procedure satisfies its objective by significantly reducing, in a large majority of cases, the error, compared it with the static induction processes as well the adaptive weighting improves efficiently and rapidly the forest performance in the first iterations of the process. The static RF has no guarantee that each tree added to the current forest ameliorates its performance, compared with the DRF. Also we notice that the new DRF has the same performance that DRF but reduce the time execution to 50%, as it is shown in figure 7.

Training data	Testing data	Forest-RI	DRF	New-DRF
a, b, c	d	89.01%	90.64%	89.56%
a, b, d	c	88.64%	91.03%	90.21%
a, c, d	b	88.03%	89.78%	90.01%
b, c, d	a	89.42%	90.06%	89.76%
Average		88.77%	90.37%	89.84%

Table 2: Recognition rate (%)

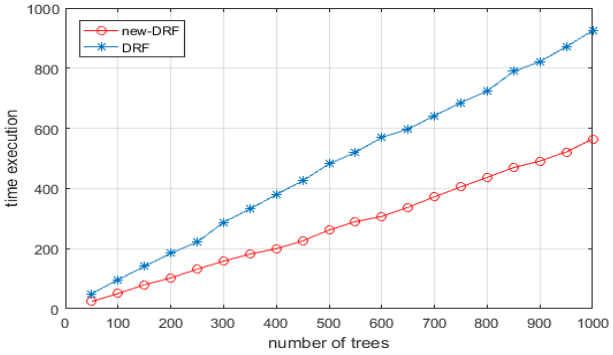

Figure 7: Time execution vs. number of trees

As it can be noted, both models have almost the same recognition rate, and we notice also that the highest rate is achieved with the class, which has the highest number of occurrences.

Word	Samples	New DRF %	DRF %
تل الغزلان (Tal al guozlan)	210	96.97	96.66
سيدي إبراهيم الزهار (sidi Ibrahim ezzahar)	210	92.45	91.33
سبعة آبار (sabaa abar)	210	95.00	95.00
الخليج (al khalij)	210	93.33	94.69
الرضاع (al radhaa)	210	94.26	95.00
نقة (nogga)	210	96.66	96.66
شعال (chaal)	210	90.33	91.45
اوتيك الجديد (autik al jidid)	142	86.66	86.66
الشوامخ (al chwamekh)	102	71.45	70.00
الشابة (al chabba)	63	26.00	26.66

Table 3: Recognition rate for some words

Table 4 illustrates a comparison of performance between some OCR systems using the database IFN/ENIT and our model. In fact, our model is standing well.

Reference	Technology	Test	recognition rate
kacem [15]	PGMs	d	91.89%
Khémiri [5]	VH-HMM	-	90.42%
Jayech [1]	MS-HMM	d	91.10-
Abandah [17]	RF	d	75.49
Kessentini [16]	HMM	d	93.04
Proposed system	Forest-RI	d	89.01%
	DRF	d	90.64%
	new-DRF	d	89.56%

Table 4: Comparison of the proposed system with the performance of other OCR systems using IFN\ENIT

6. Conclusion

An efficient Arabic handwriting recognition system based on new DRFs has been presented. Extensive experiments with various numbers of trees, characteristics and codebook size have been performed on the IFN/ENIT database. The developed new DRFs model has been performed better to the state-of-the-art models. It has been also faster than DRF, proposed by Bernard. In the future, we will investigate other feature types and training method that will close the gap on DRFs and the state-of-the-art methods on this dataset.

References

[1] K. Jayech, K, M.A. Mahjoub, N.E.B. Amara. Synchronous Multi-Stream Hidden Markov Model for offline Arabic handwriting recognition without explicit segmentation. Neurocomputing, (214): 958-971, 2016.

[2] K. Jayech, K, M.A. Mahjoub, N.E.B. Amara. Arabic Handwriting Recognition Based on Synchronous Multi-stream HMM Without Explicit Segmentation. In International Conference on Hybrid Artificial Intelligence Systems, 2015.

[3] K. Jayech, K, M.A. Mahjoub, N.E.B. Amara. Arabic handwritten word recognition based on dynamic bayesian network. Int. Arab J. Inf. Technol., 13(3):276-283, 2016.

[4] A. Khémiri, A. K. Echi, A. Belaïd, M. Elloumi. A System for Off-Line Arabic Handwritten Word Recognition Based on Bayesian Approach. In International Conference on Frontiers in Handwriting Recognition (ICFHR), 2016.

[5] A. Khémiri, A. K. Echi, A. Belaïd, M. Elloumi. Arabic handwritten words off-line recognition based on HMMs and DBNs. In 13th International Conference on Document Analysis and Recognition (ICDAR), 2015.

[6] N. Aouadi, A. K. Echi, A. Belaïd. A recognition based approach for segmenting touching components in Arabic manuscripts. In 13th International Conference on Document Analysis and Recognition (ICDAR), 2015.

[7] Y. Chherawala, P.P. Roy, M. Cheriet. Feature set evaluation for offline handwriting recognition systems: application to the recurrent neural network model. IEEE transactions on cybernetics, 46(12): 2825-2836, 2016.

[8] S. Bernard, S. Adam, L. Heutte. Using random forests for handwritten digit recognition. In Ninth International Conference on Document Analysis and Recognition, 2: 1043-1047, 2007.

[9] L. Breiman. Random Forests Statistics. Department, University of California, Berkeley, Machine Learning, 45: 5–32, 2001.

[10] H. Mohsen, H. Kurban, M. Jenne, M. Dalkilic. A new set of Random Forests with varying dynamic data reduction and voting techniques. In International Conference on Data Science and Advanced Analytics (DSAA), 2014.

[11] R. Abdeen, M. Afifi, A.B. El-Sisi. Improved Arabic handwriting word segmentation approach using Random Forests. In 12th International Conference of Computer Systems and Applications (AICCSA), 2015.

[12] M. Rashad, N.A. Semary. Isolated Printed Arabic Character Recognition Using KNN and Random Forest Tree Classifiers. In International Conference on Advanced Machine Learning Technologies and Applications, 2014.

[13] Y. Zamani, Y. Souri, H. Rashidi, S. Kasaei. Persian handwritten digit recognition by random forest and convolutional neural networks. In 9th Iranian Conference on Machine Vision and Image Processing (MVIP), 2015.

[14] S. Bernard, S. Adam, L. Heutte. Dynamic Random Forests. Pattern Recognition Letters, Elsevier, 33(12):1580-1586, 2012.

[15] A. Khémiri, A. K. Echi, A. Belaïd. A PGM-based System for Arabic Handwritten Word Recognition. Electronic Letters on Computer Vision and Image Analysis 13(3):41-62, 2014.

[16] Y. Kessentini, T. Paquet, A.B. Hamadou. Off-line handwritten word recognition using multi-stream Hidden Markov Models. Pattern Recognition Letters, 31 (1):60-70, 2010.

[17] G. Abandah, F. Jamour. Recognizing handwritten Arabic script through efficient skeleton-based grapheme segmentation algorithm. Proceedings of the International Conference on Intelligent Systems Design and Applications, 2010.

[18] S. B. Ahmed, S. Naz, M. I. Razzak, S. F. Rashid, M. Z. Afzal, and T. M.Breuel. Evaluation of cursive and non-cursive scripts using recurrentneural networks. Neural Computing and Applications, 27(3): 603–613, 2016.