An Ensemble Learning Approach using Decision Fusion for the Recognition of Arabic Handwritten Characters

Rihab Dhief[1], Rabaa Youssef[1,2] and Amel Benazza[1]
[1]*University of Carthage SUP'COM, LR11TIC01, COSIM Lab., 2083, El Ghazala, Tunisia*
[2]*INSAT, University of Carthage, Tunisia*

Keywords: Handwritten Arabic Character Recognition, Skeletonization, Freeman Chain Code, Heutte Descriptors, Feature Extraction, Supervised Machine Learning Algorithms, Deep Learning.

Abstract: The Arabic handwritten character recognition is a research challenge due to the complexity and variability of forms and writing styles of the Arabic alphabet. The current work focuses not only on reducing the complexity of the feature extraction step but also on improving the Arabic characters' classification rate. First, we lighten the preprocessing step by using a grayscale skeletonization technique easily adjustable to image noise and contrast. It is then used to extract structural features such as Freeman chain code and Heutte descriptors. Second, a new model using the fusion of results from machine learning algorithms is built and tested on two grayscale images' datasets: IFHCDB and AIA9K. The proposed approach is compared to state-of-the-art methods based on deep learning architecture and highlights a promising performance by achieving an accuracy of 97.97% and 92.91% respectively on IFHCDB and AIA9K datasets, which outperforms the classic machine learning algorithms and the deep neural network chosen architectures.

1 INTRODUCTION

Arabic is an international language widely spoken in the world. The Arabic alphabetic contains 28 letters written from right to left and they are highly similar to each other. The Arabic language has flourished in several eras, and gained a scientific, literary and religious weight. Digitizing this heritage is very important for better archiving and exploration.

Optical Character Recognition (OCR) (Borovikov, 2014) is the process of converting images of handwritten or printed text into digital, machine-editable text. Despite the attention given to the optical character recognition field and the interesting results of the literature (Alaei et al., 2012; Rajabi et al., 2012; Siddhu et al., 2019; Althobaiti and Lu, 2017; Altwaijry and Al-Turaiki, 2021; Boulid et al., 2017; KO and Poruran, 2020; Balaha et al., 2021b), the recognition of Arabic handwritten characters still has its challenges and difficulties. In fact, its cursive writing style generates a variation in shape, curve angles and size of each character, depending on its position in the word. Furthermore, various characters in the Arabic alphabet have the same main body but can be differentiated only by the position and number of the diacritics (Lutf et al., 2010). Many Arabic datasets are provided by open-access resources. Some datasets include only simplified binary characters (El-Sawy et al., 2017; Altwaijry and Al-Turaiki, 2021), while some others gather grayscale images (Torki et al., 2014; Mozaffari et al., 2006).

In this work, we propose to investigate a new architecture that brings together many algorithms in order to take advantages of each of them. First, we propose to combine structural and statistical features. To obtain the structural features, the Self-Noise and Contrast Controlled Thinning algorithm (Youssef et al., 2016) is implemented. This algorithm lighten the preprocessing step by improving the model robustness to noise and low-contrast. In fact, the SCCT algorithm has proven its efficiency in the medical field (Mallat and Youssef, 2016), when directly applied on X-Ray images. Second, the principle contribution of the current work consists of implementing a decision fusion based on the most efficient machine learning classifiers, namely SVM, KNN and RF.

The structure of this paper is described as follows. First, Section 2 reviews the pertinent works in Arabic handwritten recognition. Second, Section 3 describes the datasets used in this work. Then, Section 4 highlights the feature extraction step, using the SCCT skeletonization. Section 5 details the proposed model: the fusion of machine learning classifiers. Experimental results are showed in Section 6. Finally, conclusions are drawn in Section 7.

Dhief, R., Youssef, R. and Benazza, A.
An Ensemble Learning Approach using Decision Fusion for the Recognition of Arabic Handwritten Characters.
DOI: 10.5220/0010839500003122
In *Proceedings of the 11th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2022)*, pages 51-59
ISBN: 978-989-758-549-4; ISSN: 2184-4313
Copyright © 2022 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

2 RELATED WORK

For years, the problem of classification of Arabic characters has existed, several studies have been carried out to find as much precision as possible. Three important steps in the recognition system should be treated : Preprocessing , Feature Extraction and Classification.

2.1 Preprocessing

Preprocessing aims at removing unnecessary information without modifying the form of the object. This is a fundamental step to ensure good classification results. Traditional preprocessing methods are generally filtering and noise removal (Althobaiti and Lu, 2017; Sahlol et al., 2014). Thining and Object contour are also often used techniques especially for identifying the object structure (Boufenar et al., 2018). These techniques are efficient in the character recognition field but since they are based on binary image, they still risk losing useful information during the binarization step.

2.2 Feature Extraction

Feature engineering is a primordial step for every ML learning model. It consists of transforming image data into features that better represent the underlying problem to the predictive models, resulting in improved model accuracy on unseen data. Recently, many researchers use deep features for the feature extraction step. This method does not need further preprocessing and achieves high performance according to the literature (Altwaijry and Al-Turaiki, 2021; Balaha et al., 2021a). Despite its effectiveness, it requires a higher computational costs than traditional methods. Traditional feature extraction is performed through two main shape description approaches: statistical and structural. Studies based on supervised learning using only structural features (Althobaiti and Lu, 2017), or statistical approaches (Alaei et al., 2012; Rajabi et al., 2012), and also the combination of both (Alaei et al., 2012; Zanchettin et al., 2012; Sahlol et al., 2014; Sahlol et al., 2016; Boufenar et al., 2018; Siddhu et al., 2019) were proposed in the literature with the latter achieving more interesting results. Combining both type of features explains the need for merging different classifiers since each classifier needs different form of features for better results.

2.3 Classification

Previous works propose two main streams of approaches to deal with the Arabic handwritten recognition problem, namely Deep Neural Network (DNN) architectures (Altwaijry and Al-Turaiki, 2021; KO and Poruran, 2020; Balaha et al., 2021a) and classical Machine Learning (ML) techniques (Zanchettin et al., 2012; Alaei et al., 2012; Rajabi et al., 2012; Sahlol et al., 2014; Sahlol et al., 2016; Boufenar et al., 2018; Siddhu et al., 2019; Ali et al., 2020). On the one hand, and regarding the use of Deep Neural Networks, authors of (Altwaijry and Al-Turaiki, 2021) propose a convolutional neural network approach (CNN) for the recognition of Arabic handwritten characters using a small binary dataset and achieving an accuracy result of 97%, while authors of (Boulid et al., 2017; KO and Poruran, 2020; Balaha et al., 2021a) conduct a research on grayscale character images with a result of 96%. On the other hand, and regarding the use of classical Machine Learning (ML) approaches, three main models have been widely applied, namely Support Vector Machine (SVM) (Zanchettin et al., 2012; Alaei et al., 2012; Rajabi et al., 2012; Sahlol et al., 2014; Siddhu et al., 2019), Random Forest (RF) (Sahlol et al., 2016; Rashad and Semary, 2014) and K-Nearest Neighbors (KNN) (Zanchettin et al., 2012; Rajabi et al., 2012; Sahlol et al., 2014; Boufenar et al., 2018), with a highlight on SVM results in most of the cited experimental results. Furthermore, the idea of combining multiple classifiers emerged in the past few years using both DL (Bosowski et al., 2021) and ML algorithms (Zhao and Liu, 2020; Kaoudja et al., 2019). Ensemble learning is an efficient way to take advantage of different classifiers especially when they have heterogeneous inputs (different features type). In fact, the authors of (Zanchettin et al., 2012) combine the SVM and KNN classifiers, also, RF and KNN combination is proposed in (Zhao and Liu, 2020) for numeral recognition, and multi-classifier system for Arabic calligraphy recognition is built in (Kaoudja et al., 2019), merging three classifiers namely: Multilayer Perceptron (MLP), SVM, and KNN. Although state-of-the-art methods for recognising binary Arab handwritten characters have achieved satisfactory results(Alaei et al., 2012; Rajabi et al., 2012; Zanchettin et al., 2012; Sahlol et al., 2014; Zhao and Liu, 2020; Kaoudja et al., 2019), further improvements remain conceivable by adopting new approaches and methodologies on grayscale image datasets.

3 DATASETS

Two datsets are used in the curret work, namely Isolated Farsi Handwritten Character Data Base (IFHCDB) and AlexU Isolated Alphabet (AIA9K). Both datasets contain grayscale images.

3.1 The IFHCDB Dataset

The IFHCDB dataset (Mozaffari et al., 2006) includes isolated Farsi and Arabic handwritten characters. In our project, only Arabic letters are considered. The Arabic alphabet contains 28 letters and thus, our model contains 28 classes. The main issue with this dataset is the lack of balance between the classes associated to each character. In total, IFHCDB dataset contains 51029 Arabic character with different number of observation varies from 40 to 10000 per class. To reduce this unbalance, we merge classes based on same character body and variable diacritics. We end up with 18 classes presented in Figure 1.

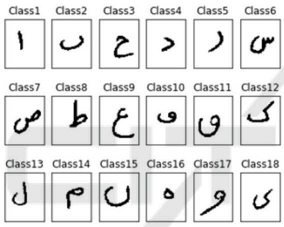

Figure 1: The character bodies.

3.2 The AIA9K Dataset

The total size of AIA9K dataset (Torki et al., 2014) is 8736 divided in 28 classes with number of observations varies from 251 and 278 per class. For sake of clarity, we merged classes with same character body, as for IFHCDB dataset.

4 HANDCRAFTED FEATURE EXTRACTION

Two types of features are extracted: structural ones from the skeleton and statistical ones from the character body. Heutte et al. (Heutte et al., 1998) collected a set of statistical and structural features that describes the character globally (projection, moments and profiles) and locally (Intersection with straight lines, holes, concave arcs, junctions, endpoints and extrema). These descriptors browse almost all what could characterize a character. We briefly describe these features in the next two subsections.

4.1 Features Extracted from the Character Body

Hu Moments: The seven equations are detailed in (Hu, 1962) and are invariant to position, size and orientation of the character.

Projections: Vertical and horizontal projections derived from the histograms of the character image are calculated and the maxima from each is extracted to locate the most significant object pixels number vertically and horizontally.

Profiles: The profile correspond to the set of differences between two consecutive pixels (between two ordinates in the right and left profiles or two abscissas otherwise). The profiles provide information about the harmony of the character.

Heights and Widths: They describe the character in terms of height (respectively width) in particular locations (1/5, 1/2 and 4/5). They are extracted from the difference between the left and the right raw profiles (respectively bottom and top) of the character bounding box.

Extrema: Top, bottom, left and right extrema of the character are extracted by browsing the image from left to right, right to left, top to bottom and bottom to top. Each time, the first pixel which does not have an 8-connected object neighbour is recorded.

Concave Arcs: They are extracted from the object contours. A concave arc is the set of three consecutive points that form an angle of less than 180 degrees.

Ratio: The height to width ratio is extracted and aims at characterizing the spread of the character.

4.2 Features Extracted from the Skeleton

Self-Noise and Contrast Controlled Thinning (SCCT) was developed by the authors of (Youssef et al., 2016). It generates a smooth silhouette using two thinning parameters: contrast and noise. This study proposes to relax the topology preservation property of homotopic thinning by considering local noise and contrast, as shown by Figure 2. Applying this skeletonization method improves the skeleton's quality compared to binary skeletonization (Zhang and

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

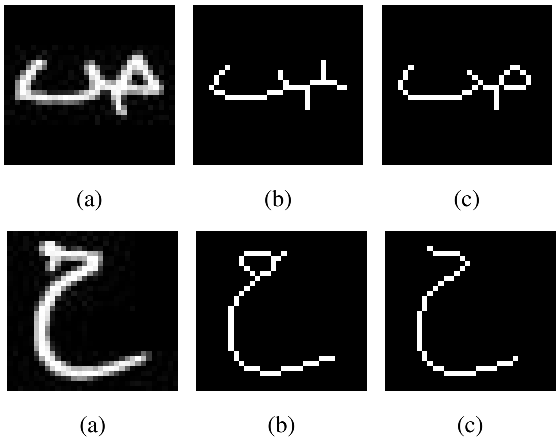

Figure 2: Comparison:(a) Character body, (b) Binary thinning, (c) Grayscale thining.

Suen, 1984) and consequently, the final classification results.

Figure 2 presents two examples from class 14 and 6. The binary skeletonization (Zhang and Suen, 1984) fails at recognizing the character's hole for the letter "sad" while creating one in the wrong place for letter "ha". This confusion is problematic since hole detection remains an essential feature to separate classes.

Endpoints and Junctions: An endpoint is an object pixel that has only one 8-connected object neighbor. A junction is defined as a pixel having at least three 8-connected object neighbors that separate the background into three or more 4-connected components.

Holes and Intersections with Straight Lines: A hole is detected when the image background contains more than one 4-connected component. The intersections extraction are defined as follows: two horizontal lines (1/3 height and 2/3 height) and a vertical line crossing the character's centre of gravity.

Freeman Chain Code: Freeman's chain converts a skeleton image into a directional code. Our process of extracting the Freeman chain code is described as follows: the starting point is the first encountered endpoint pixel. Then, based on the position of the current pixel neighbour, we pick the appropriate Freeman direction to start constructing the chain code. Figure 3 (a) states the common choice of the 8 directions. An example in Figure 3 (b) details the Freeman chain code constructed for letter "Lam".

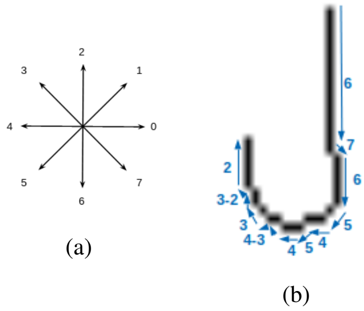

Figure 3: (a) Freeman 8 directions, (b) The chain code of the letter "lam".

4.3 Features Extracted from Diacritics

Diacritics are obtained after removing the character body from the image. The number, size and position of each diacritic are extracted to differentiate between characters having similar bodies but differing in diacritics. Figure 4 shows some example of diacritics encountered in the AIA9K dataset.

Figure 4: Different types of diacritics.

5 THE PROPOSED METHODOLOGY: ENSEMBLE LEARNING CLASSIFIER USING DECISION FUSION

Once these statistical and morphological features are computed, they should undergo a classification in order to recognize the underlying character. In this respect, we adopt a decision fusion strategy based on the most efficient machine learning classifiers used in the character recognition field, namely weighted SVM, weighted RF and K-NN (Ayodele, 2010).

5.1 The Classifiers

Weighted Support Vector Machine: SVM is initially used for binary classification. It consists of defining a hyperplane that separates two classes. As we are faced to a multi-classification problem with $m > 2$ classes, the strategy one versus one is adopted in order to apply $m(m-1)/2$ binary classifiers. Weighted SVM is adopted in our work because of the datasets unbalance.

Weighted Random Forest: Random forest consists of a set of decision trees. Every decision tree gives a

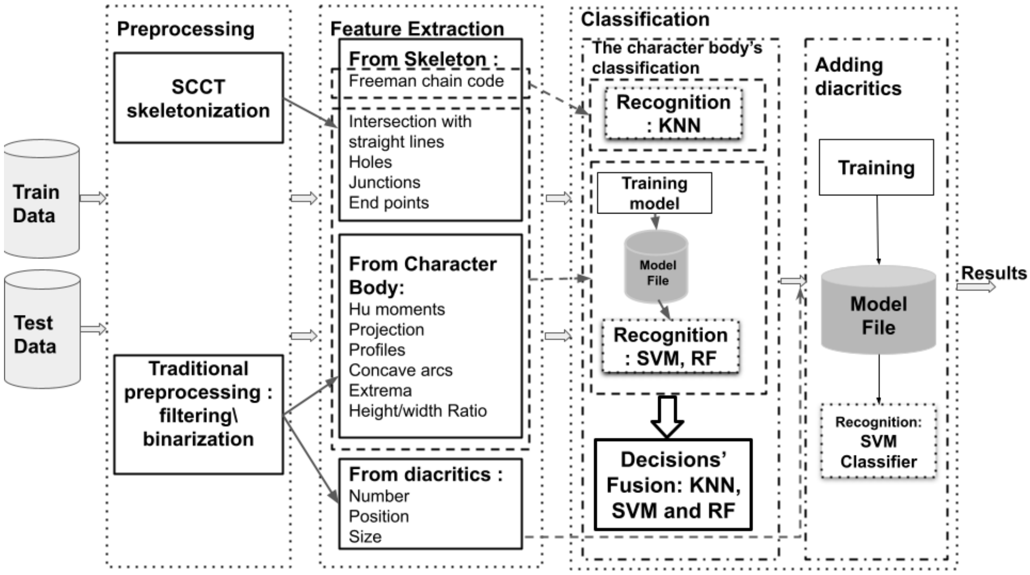

Figure 5: The proposed architecture for Arabic handwritten characters recognition.

predicted class. Then, the most frequently predicted classes is chosen. RF and SVM are used to classify characters using numeric statistical and structural features.

K-Nearest Neighbors: K-NN is essentially based on the calculation of metrics (distance) between observations. For each new observation, we can predict its class by looking at the classes of its nearest neighbors. The number of considered neighbors in the K-NN classifier is the parameter K which is set empirically from the beginning. The K-NN classifies the Freeman chains by using the Levenshtein distance (Levenshtein et al., 1966), which is a string metric for measuring the difference between two sequences.

5.2 Ensemble Learning: Decision-fusion Principle

In this section, the suggested methodology for the recognition of Arabic handwritten characters is presented. Figure 5 illustrates the proposed system in a block diagram. First, a preprocessing step is presented, where two information sources of the character are described: the binary character itself and its skeleton graph. The aim of this work is to test the SCCT method, which makes it possible to avoid part of the pretreatments, and to see its effect on the classification rate.The second step consists in extracting features from each character form. Structural features are extracted from the skeleton, while statistical ones are derived from the character body. Besides, the number, position and size are also calculated for the diacritics.

The final step is the classification, which is implemented in two main steps :

1. First stage: The character body classification: As described in subsection 5.1, three main classifiers are used, namely SVM, RF, K-NN and for which we choose different features as input. In fact, K-NN uses only the Freeman chain code feature since this chain requires the use of a specific metric which is here the Levenshtein distance. Regarding SVM and RF, we choose to implement them using the remaining features described in this work. Since the classification error is different from a classifier to another, a comparison between their results is made, and a vote that merges the respective decisions is built. In fact, we choose the most common predicted class between the three classification results. If the three predicted classes are different, the class corresponding to the best global accuracy is picked.
2. Second stage: Separate merged classes using the diacritics: in fact, due to the similarity between classes and the notable unbalance of the datasets, merging classes having the same body but differing by the diacritics' forms was adopted to improve the global accuracy. In this step, we separate the merged classes using information related to the diacritics. Each character is classified by adding the features related to its diacritics, and in this case, the SVM classifier is used since it generates better results when dealing with unbalanced data.

Table 1: Comparison between binary and grayscale methods using IFHCDB dataset.

SCCT	Thome	Zhang
93.6%	92.9%	93.3%

Table 2: Parameter tuning for the three classifiers.

	Weighted SVM	Weighted RF	KNN
	Kernel = Linear	Max−depth = 50	Distance=Levenshtein(Levenshtein et al., 1966)
IFHCDB	C = 16	Number of estimators = 900	K = 5
AIA9K	C=12	Number of estimators = 900	K = 7

6 EXPERIMENTAL RESULTS

Our contribution lies in the use of the SCCT skeletonization and the decision fusion of the three classifiers. Thus, in this section, we evaluate the results of the two contributions separately.

6.1 SCCT Contribution

In this part, a comparison between binary and grayscale skeletonization is made on the IFHCDB database. Since the Freeman chain code is the descriptor that fully exploits the skeleton: its shape, endpoints and junctions, we choose to use it for testing the efficiency of the SCCT skeletonization and thus to implement the K-NN classifier. Alongside the SCCT, two binary methods are used in the comparison : Thome skeletonization (Merad et al., 2010) and Zhang skeletonization (Zhang and Suen, 1984). The adjustment of the SCCT parameters regarding the contrast and noise of the image aims at finding a compromise between preserving the topology of the object and removing noise related information. For this purpose, the two following parameters must be set in the SCCT skeletonization technique:

- The standard deviation of the background noise: Since all the images in the dataset are acquired under the same conditions, we can precalculate noise standard deviation empirically by choosing a region from a set of image background, on which we calculate the standard deviation.
- Test confidence level: This parameter is intimately linked to contrast level. According to the authors (Youssef et al., 2016), and for images that are correctly contrasted, a confidence level of 0.01 is adequate.

According to Table 1, the results provided by SCCT skeletonization comfort us in our choice, since we can remove all preprocessing steps and, at the same time, improve classification results. By doing so, the risk of damaging significant information is reduced. These results support our first observations in Figure 2.

6.2 Classifiers Decision Fusion Contribution

Parameter Tuning: Cross-validation was used to compare the performance of different predictive models: weighted SVM, weighted RF and K-NN. A 5-fold cross-validation was conducted on the training dataset in order to choose the best parameters configuration. Table 2 details the chosen parameters values for each model on respective dataset. All parameters were chosen empirically.

Results: The global accuracy for the three classifiers are presented in Table 3. Concerning the IFHCDB dataset, we need to further merge class ”ba” and ”fa” since we notice an important confusion between the two classes: 18% from class ”fa” were predicted belonging to class ”ba”. In addition, we also merge classes 2 and 15, classes 10 and 11, in the AIA9K data for the same reason.
Globally, the IFHCDB dataset results are better than those of the AIA9k database due to size issue. IFHCDB dataset contains 50k images while AIA9K presents only 8k images. According to Table 3, weighted SVM works better with unbalanced data, while RF gives higher results with small data. KNN has the lowest accuracy, but provides different information since it uses another type of feature, the ’Freeman chain code’.

Table 3: Accuracy of different classifiers.

Models	IFHCDB	AIA9K
SVM	**97.88%**	87.56%
RF	96.92%	**91.08%**
KNN	94.43%	84.83%

Table 4: Accuracy results on IFHCDB and AIA9K dataset of separate classifiers compared to the decision fusion approach.

	IFHCDB		AIA9K	
	Class ”ha”	Class ”ain”	Class ”ba+noun”	Class ”kef”
KNN	97%	**94%**	**93%**	62%
Weighted SVM	99%	91%	90%	**79%**
Weighted RF	**100%**	93%	88%	77%
Fusion	**100%**	**96%**	92%	**89%**

Table 5: Accuracy compared to the literature.

Papers	Classifier	Database	Number of classes	Accuracy
(Alaei et al., 2012)	SVM	IFHCDB	32-classes	96.91%
(Boulid et al., 2017)	Neural Network	IFHCDB	28-classes	96%
(KO and Poruran, 2020)	CNN	IFHCDB	28-classes	96.3%
Proposed model	Fusion of classifiers' decisions	IFHCDB	28-classes	**97.97%**
(Balaha et al., 2021a)	Deep learning system	AIA9K	28-classes	93.3%
Proposed model	Fusion of classifiers' decisions	AIA9K	28-classes	92.91%

In fact, and according to results exposed in Table 4 we notice that each classifier succeeds/fails in different situations. For example, in the case of the IFHCDB dataset, the global accuracy of K-NN is more interesting on letter ”ain”, while RF achieves 100% on letter ”ha”. In the case of the AIA9K dataset, same remark can be made since we notice a 10% improvement in classification result when using fusion on class ”kef”, compared to the best classifier (Weighted SVM). This reveals that the classification error is different from a classifier to another and thus, support the idea of merging the decisions of the three machine learning approaches.

The proposed decision fusion contribution has improved the accuracy up to 98.74% for the IFHCDB dataset, which is a significant result. The accuracy rate of different classes varies from 85% to 100%. Almost all the classes have an accuracy greater than 94%. And, a F_1-score of 97.5% is obtained. For the AIA9K dataset, the fusion result gives an accuracy of 94.58% and an F_1-score of 94.5%, which are interesting results regarding the limited size of the dataset.

In order to process the 28 classes, we moved to the second stage where features related to the diacritics are used: number, size, and position. The Weighted SVM classifier is implemented to separate similar character bodies. In this step, the separation is done with 100% of precision in most classes. However, there is an issue with very similar characters' diacritics, such as letter 'ba' and letter 'tha', leading to a small decrease in overall accuracy : 97.97% for IFHCDB and 92.91% for AIA9K.

Comparison with State of the Art: Table 5 presents our results and the ones of previously cited works, for instance, deep learning models (Boulid et al., 2017; KO and Poruran, 2020) and SVM classifier of (Alaei et al., 2012). Since we performed the classification on the same datasets as the above cited works, we use in this comparison the results detailed in their respective papers.

By mixing structural and statistical features extracted from both the character body and the skeleton, and by combining traditional classifiers to bring out the best of each, we obtain the highest accuracy of 97.97% among the cited methods from literature. Another important remark is that these results are obtained on highly unbalanced dataset without using any data augmentation technique. One can also notice that the separation of merged classes in the case of AIA9K dataset did not benefit the overall accuracy, since some of the separated classes has really few observations, which declined the overall accuracy.

7 CONCLUSIONS

In this paper, a method to recognize handwritten Arabic character is presented. The proposed approach includes a data analysis step to extract each character's most accurate descriptors and a classification step. In the first step, a new technique of skeletonization is used to improve the feature extraction phase. In the modelling step, a new classification method is proposed resulting in an interesting accuracy rate compared to separate classifiers and Deep Learning architectures when tested on the IFHCDB dataset. A data augmentation technique should be done in future works to improve the result on AIA9K dataset due to its small size.

REFERENCES

Alaei, A., Pal, U., and Nagabhushan, P. (2012). A comparative study of persian/arabic handwritten character recognition. In *2012 International Conference on Frontiers in Handwriting Recognition*, pages 123–128. IEEE.

Ali, A. A. A., Suresha, M., and Ahmed, H. A. M. (2020). A survey on arabic handwritten character recognition. *SN Computer Science*, 1(3):1–10.

Althobaiti, H. and Lu, C. (2017). A survey on arabic optical character recognition and an isolated handwritten arabic character recognition algorithm using encoded freeman chain code. In *2017 51st Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6. IEEE.

Altwaijry, N. and Al-Turaiki, I. (2021). Arabic handwriting recognition system using convolutional neural network. *Neural Computing and Applications*, 33(7):2249–2261.

Ayodele, T. O. (2010). Types of machine learning algorithms. *New advances in machine learning*, 3:19–48.

Balaha, H. M., Ali, H. A., Saraya, M., and Badawy, M. (2021a). A new arabic handwritten character recognition deep learning system (ahcr-dls). *Neural Computing and Applications*, 33(11):6325–6367.

Balaha, H. M., Ali, H. A., Youssef, E. K., Elsayed, A. E., Samak, R. A., Abdelhaleem, M. S., Tolba, M. M., Shehata, M. R., Mahmoud, M. R., Abdelhameed, M. M., et al. (2021b). Recognizing arabic handwritten characters using deep learning and genetic algorithms. *Multimedia Tools and Applications*, pages 1–37.

Borovikov, E. (2014). A survey of modern optical character recognition techniques. *arXiv preprint arXiv:1412.4183*.

Bosowski, P., Bosowska, J., and Nalepa, J. (2021). Evolving deep ensembles for detecting covid-19 in chest x-rays. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 3772–3776. IEEE.

Boufenar, C., Batouche, M., and Schoenauer, M. (2018). An artificial immune system for offline isolated handwritten arabic character recognition. *Evolving Systems*, 9(1):25–41.

Boulid, Y., Souhar, A., and Elkettani, M. Y. (2017). Handwritten character recognition based on the specificity and the singularity of the arabic language. *International Journal of Interactive Multimedia & Artificial Intelligence*, 4(4).

El-Sawy, A., Loey, M., and El-Bakry, H. (2017). Arabic handwritten characters recognition using convolutional neural network. *WSEAS Transactions on Computer Research*, 5:11–19.

Heutte, L., Paquet, T., Moreau, J.-V., Lecourtier, Y., and Olivier, C. (1998). A structural/statistical feature based vector for handwritten character recognition. *Pattern recognition letters*, 19(7):629–641.

Hu, M.-K. (1962). Visual pattern recognition by moment invariants. *IRE transactions on information theory*, 8(2):179–187.

Kaoudja, Z., Kherfi, M. L., and Khaldi, B. (2019). An efficient multiple-classifier system for arabic calligraphy style recognition. In *2019 International Conference on Networking and Advanced Systems (ICNAS)*, pages 1–5. IEEE.

KO, M. A. and Poruran, S. (2020). Ocr-nets: Variants of pre-trained cnn for urdu handwritten character recognition via transfer learning. *Procedia Computer Science*, 171:2294–2301.

Levenshtein, V. I. et al. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.

Lutf, M., You, X., and Li, H. (2010). Offline arabic handwriting identification using language diacritics. In *2010 20th International Conference on Pattern Recognition*, pages 1912–1915. IEEE.

Mallat, K. and Youssef, R. (2016). Adaptive morphological closing based on inertia tensor for structuring element estimation. In *2016 International Symposium on Signal, Image, Video and Communications (ISIVC)*, pages 253–258. IEEE.

Merad, D., Aziz, K.-E., and Thome, N. (2010). Fast people counting using head detection from skeleton graph. In *2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 233–240. IEEE.

Mozaffari, S., Faez, K., Faradji, F., Ziaratban, M., and Golzan, S. M. (2006). A comprehensive isolated farsi/arabic character database for handwritten ocr research. In *Tenth International Workshop on Frontiers in Handwriting Recognition*. Suvisoft.

Rajabi, M., Nematbakhsh, N., and Monadjemi, S. A. (2012). A new decision tree for recognition of persian handwritten characters. *International Journal of Computer Applications*, 44(6):52–58.

Rashad, M. and Semary, N. A. (2014). Isolated printed arabic character recognition using knn and random forest tree classifiers. In *International Conference on Advanced Machine Learning Technologies and Applications*, pages 11–17. Springer.

Sahlol, A., Abd Elfattah, M., Suen, C. Y., and Hassanien, A. E. (2016). Particle swarm optimization with random forests for handwritten arabic recognition system. In *International Conference on Advanced Intelligent Systems and Informatics*, pages 437–446. Springer.

Sahlol, A. T., Suen, C. Y., Elbasyoni, M. R., and Sallam, A. A. (2014). Investigating of preprocessing techniques and novel features in recognition of handwritten arabic characters. In *IAPR Workshop on Artificial Neural Networks in Pattern Recognition*, pages 264–276. Springer.

Siddhu, M. K., Parvez, M. T., and Yaakob, S. N. (2019). Combining statistical and structural approaches for arabic handwriting recognition. In *2019 International conference on computer and information sciences (ICCIS)*, pages 1–6. IEEE.

Torki, M., Hussein, M. E., Elsallamy, A., Fayyaz, M., and Yaser, S. (2014). Window-based descriptors for arabic handwritten alphabet recognition: a comparative study on a novel dataset. *arXiv preprint arXiv:1411.3519*.

Youssef, R., Sevestre-Ghalila, S., Ricordeau, A., and Benazza, A. (2016). Self noise and contrast controlled thinning of gray images. *Pattern Recognition*, 57:97–114.

Zanchettin, C., Bezerra, B. L. D., and Azevedo, W. W. (2012). A knn-svm hybrid model for cursive handwriting recognition. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Zhang, T. Y. and Suen, C. Y. (1984). A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 27(3):236–239.

Zhao, H.-h. and Liu, H. (2020). Multiple classifiers fusion and cnn feature extraction for handwritten digits recognition. *Granular Computing*, 5(3):411–418.

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS