西 南 交 通 大 学 学 报

第 55 卷 第 1 期
2020 年 2 月

JOURNAL OF SOUTHWEST JIAOTONG UNIVERSITY

Vol. 55 No. 1
Feb. 2020

ISSN: 0258-2724

DOI：10.35741/issn.0258-2724.55.1.12

Research Article

Computer Science

MULTI-FONT ARABIC ISOLATED CHARACTER RECOGNITION USING COMBINING MACHINE LEARNING CLASSIFIERS

结合机器学习分类器的多字体阿拉伯语隔离字符识别

Raidah S. Khudeyer [a], Maytham Alabbas [b,] *, Mustafa Radif [c]
[a] Department of CIS, College of CSIT, University of Basrah
Al-Ashar–Corniche St., Basrah, Iraq, raidah.khudayer@uobasrah.edu.iq
[b] Department of CS, College of CSIT, University of Basrah
Al-Ashar–Corniche St., Basrah, Iraq, ma@uobasrah.edu.iq
[c] Department of IS, College of CSIT, University of Al-Qadisiyah
P.O. Box 88, Al Diwaniyah, Al-Qadisiyah, Iraq, mustafa.radif@qu.edu.iq

Abstract

Nowadays, optical character recognition is one of the most successful automatic pattern recognition applications. Many works have been done regarding the identification of Latin and Chinese characters. However, the reason for having few investigations for the recognition of Arabic characters is the complexity and difficulty of Arabic characters identification compared to the others. In the current work, we investigate combining multiple machine learning algorithms for multi-font Arabic isolated characters recognition, where imperfect and dimensionally variable input charactersare faced. To the best of our knowledge, there is no such work yet available in this regard. Experimental results show that combined multiple classifiers can outperform each individual classifier produces by itself. The current findings are encouraging and opens the door for further research tasks in this direction.

Keywords: Character Recognition, Generalized Regression Neural Network, Support Vector Machine, Multi-Font Recognition, System Combination

摘要 如今，光学字符识别是最成功的自动模式识别应用程序之一。关于识别拉丁文和中文字符已经完成了许多工作。然而，很少进行识别阿拉伯字符的研究的原因是与其他字符相比，阿拉伯字符识别的复杂性和困难性。在当前的工作中，我们研究了将多种机器学习算法结合在一起用于多字体阿拉伯语孤立字符识别的问题，其中面对不完美和尺寸可变的输入字符。据我们所知，在这方面尚无此类工作。 实验结果表明，组合的多个分类器可以胜过每个单独的分类器本身产生的性能。当前的发现令人鼓舞，并为进一步研究该方向打开了大门。

关键词: 字符识别，广义回归神经网络，支持向量机，多字体识别，系统组合

I. INTRODUCTION

Over 300 million people around the world speak Arabic. Arabic script recognition is a significant tool in the development of new communication means. It presents a crucial point in human-machine communication development [1]. Arabic belongs to the family of cursive scripts. Over 450 fonts for printed Arabic characters, which have different sizes, generate a large number of morphological varieties for these characters [2]. The printed Arabic script recognition, therefore, remains a significant challenge. Optical character recognition (OCR) is the process of translating handwritten images, typewritten images, or printed text into a format understood by machines for the purpose of editing, indexing/searching, and a storage size reduction [3], [22]. Arabic OCR is a challenge due to difficulties of Arabic characters that arise from the characters' characteristics and how to connect and write them. Considering these reasons, the multi-font Arabic characters recognition, which is a classification of a very large number of pattern classes, becomes a much harder problem. Considerable attention has been paid to multi-font Latin [3], [4], [5] and Chinese [6], [7] character recognition, while for Arabic it is still limited due to the mentioned challenge.

There are a few researches that address the multi-font character recognition for Arabic. Amor and Amara [8] presented a hybrid technique based on both hidden Markov models (HMMs) and artificial neural networks (ANNs), with Hough transform for features extraction, for classification multi-font Arabic characters. Khorsheed [9] has developed an Arabic multi-font OCR system using discrete HMMs along with intensity-based features. The author implemented character models using mono and tri models. Kilic et al. [10] proposed an OCR system which implements segmentation, normalization, edge detection and recognition of the Ottoman script, which is a version of the Perso-Arabic alphabet. The authors use Cellular Neural Network (CNN) as edge detection of Ottoman scripts, while the Support Vector Machine (SVM) model is employed for recognizing Ottoman characters. Izakian et al. [11] proposed a chain code-based approach along with other significant peculiarities such as the number and location of dots and auxiliary parts, and the number of holes existing in the isolated character for identification of Farsi/Arabic characters. Mazroui and Elmiad [2] developed an approach based on the properties of Bézier curves for classification. The tests were carried out on a very large sample of 23 fonts.

This paper has five sections. Following this introduction, Section 2 presents the aim of the current work. Section 3 analyzes the Arabic characters with short description to the main problems associated with them. Section 4 gives a brief explanation of GRNN, kNN, and SVM. Section 5 describes the proposed technique. Section 6 compare the results of the current system and the others. Finally, Section 7 provides the conclusion of this paper.

II. RESEARCH AIM

In this paper, we aim at combining multiple machine learning classifiers for the recognition of multi-font isolated printed Arabic characters. This technique based on combining three machine learning classifiers (generalized regression neural network (GRNN), k-nearest neighbors (kNN), and support vector machine (SVM)) by majority vote and backoff to structural similarity index (SSIM) to produce the final decision. Simulation results have demonstrated that the combination techniques always outperform each technique in isolation in terms of recognition rate.Systems combination has been vastly applied in various fields of computer science such as pattern recognition [12] and natural language processing (NLP) [13], [14], [15].

III. ARABIC CHARACTER FEATURES

Both Arabic characters and writing have various features that make the identification of Arabic character more complicated than the character identification for other languages such as Latin and Chinese. Arabic script (in both typewritten and handwritten forms) is cursively written from right to left [16]. Below are the main Arabic character features.

1. There are 29 characters in the Arabic alphabet. These characters vary in shape depending on their position within a word. Characters can have up to four distinct forms corresponding to an initial, middle, final, or isolated position, which increase the patterns from 29 to about 111 patterns [17]. Table 1 illustrates these patterns (each cell in the table represents the multiple forms for each character depending on the position of this character in the word).

Table 1.
The forms of Arabic characters

Meem	Ayn	Seen	HHa	Hamza
مـ ـمـ ـم م	عـ ـعـ ـع ع	سـ ـسـ ـس س	حـ ـحـ ـح ح	ء ئـ ئ ى
Noon	Ghayn	Sheen	Khaa	Alif
نـ ـنـ ـن ن	غـ ـغـ ـغ غ	شـ ـشـ ـش ش	خـ ـخـ ـخ خ	ا ـا
Ha	Faa	Saad	Daal	Baa
هـ ـهـ ـه ه	فـ ـفـ ـف ف	صـ ـصـ ـص ص	د ـد	بـ ـبـ ـب ب
Waaw	Qaaf	Dhad	Dhaal	Taa
و ـو	قـ ـقـ ـق ق	ضـ ـضـ ـض ض	ذ ـذ	تـ ـتـ ـت ت
Yaa	Kaaf	Taa	Raa	Thaa
يـ ـيـ ـي ي	كـ ـكـ ـك ك	ط ـط	ر ـر	ثـ ـثـ ـث ث
	Laam	Dhaa	Zaay	Jeem
	لـ ـلـ ـل ل	ظ ـظ	ز ـز	جـ ـجـ ـج ج

2. Seventeen characters out of twenty-nine have an integral part that is associated with the character body. This integral part possibly single dot, double dots, triple dots, or zigzag (ء) that maybe placed above (ف، ق، ث، ئـ), below (ب، ي), or inside (ج) the character body [16], [17].

3. Many character groups that have the identical basic shape, but they are differentiated by the number of the dots (ب،ت،ث), the dots' position (ج،خ), or whether it is dot or zigzag (نـ، ئـ) [17].

4. There are no distinct upper and lower case letterforms.

IV. Recognition Techniques

Three different machine learning approaches were investigated to recognize multi-font isolated printed Arabic characters (Table 2). The functional description of these techniques is introduced in the following lines.

Table 2.
Pattern names and numbers

Char order	Char name	Number of patterns	Total patterns	Char order	Char name	Number of patterns	Total patterns
1	Alif	2	10	15	Dhad	4	20
2	Baa	4	20	16	Taa	4	20
3	Taa	4	20	17	Dhaa	4	20
4	Thaa	4	20	18	Ayn	4	20
5	Jeem	4	20	19	Ghayn	4	20
6	HHa	4	20	20	Faa	4	20
7	Khaa	4	20	21	Qaaf	4	20
8	Daal	2	10	22	Kaaf	4	20
9	Dhaal	2	10	23	Laam	4	20
10	Raa	2	10	24	Meem	4	20
11	Zaay	2	10	25	Noon	4	20
12	Seen	4	20	26	Ha	4	20
13	Sheen	4	20	27	Waaw	2	10
14	Saad	4	20	28	Yaa	4	20
Total number of patterns = 100							
Total patterns for 5-font = 500							

- *GRNN* is a neural network. It predicts the output of a given input data. It starts by preparing a training dataset. Training data should contain input-output mapping. After network training, new testing dataset is fed. Then its output is estimated using weighted average of the outputs of the training dataset. The used weight is calculated using the Euclidean distance between the training data and test data [18]. If the weight or distance is large, then the weight will be very small, otherwise, it will put more weight to the output. GRNN contains four basic layers. First, the input layer that feeds the input to the next layer. Second, the pattern layer that calculates the Euclidean distance and activation function. Third, the summation layer has two subparts: numerator part and denominator part. Numerator part contains summation of the multiplication of training data and activation function, while denominator part is the summation of all activation functions. This layer feeds both the numerator and denominator to the next output layer, which contains one neuron. The output is calculated by dividing the numerator part by the denominator part.
- *kNN* is one of simplest supervised machine learning algorithms used for classification problems. 'k' here is the number of nearest neighbors to examine [19].

To find nearest neighbor of unknown data, we calculate Euclidean distance from all vectors in data points, then we select the k neighbor item in dataset, which are closest to new data.

This algorithm starts with determining the value of k, then calculating the distances between the test dataset and all the training dataset. The most commonly used method for calculating distance is Euclidean**.** After that, we sort the distance and determine k nearest neighbors based on minimum distance values, then analyze the category of those neighbors and assign the category for the test data based on majority vote that gives us the predicted class.

- *SVM* is supervised learning algorithm used for classification task by separating data using hyperplanes [20]. The objective of this algorithm is to find a hyperplane in an N-dimensional space, which represents a number of features that has the maximum distance between data points of classes in problem. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.
- To build SVM, we must know that support vectors are data points which are closer to the hyperplane and influence the hyperplane position and the hyperplane orientation. Using these support vectors, we maximize the classifier margin. Deleting the support vectors will change the hyperplane position.

V. RESEARCH METHOD

The current work aims at investigating the evaluation of combining three efficient machine learning techniques (GRNN, kNN, and SVM) to classify multi-font isolated printed Arabic characters, where the given letters are imperfect with different levels of noise and dimensionally variant. Then, the results of system combination are compared with each individual technique.

Each system proceeds in three distinct stages, as follows: (i) stage one prepares the given characters image by acquisition, digitizing of the character, remove the noise, binarized and thinning; (ii) stage two extracts the distinct features of the characters; and (iii) stage three processes the extracted features to recognize the given characters. Four different systems are used through stage three, which are GRNN, kNN, SVM, and combination system among them. Figure 1 shows the block diagram of the current system.

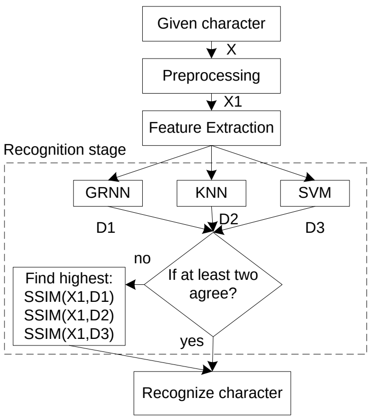

Figure 1. Block diagram of the current system

The following lines describe each stage briefly.

A. Preprocessing Stage

This stage attempts to eliminate some variability due to font style, the environment of writing, acquisition, and image digitizing. Here, three operations were used:

1) Noise Reduction

An adaptive median filter is applied here to reduce the noise in the extracted image. This filter can eliminate some of the noise with keeping the image edges.

2) Binarizing

This part is responsible for converting the given image f(x,y) into a binary image b(x,y). It is done by replacing all pixels in the given image with luminance greater than a specific threshold T with the value 1 (White); otherwise replacing the other pixels the value 0 (Black) as shown in Eq. 1.

$$b(x,y) = \begin{cases} 1 & \text{for } f(x,y) > T \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

3) Thinning

It is done to make the characters around one pixel wide.

B. Feature Extraction

The acquired characters from the given image have dissimilar dimensions (e.g., the character width of Alif (ا) is distinct from the width of Taa (ت), and the height is different as well). To treat this challenge, we used the discrete cosine transform (DCT) [21] with 64 coefficients to extract the Arabic character features. DCT is a method to convert the data of the current character image into its elementary frequency components where high-value coefficients are clustered in the upper left corner and low-value coefficients are clustered in the bottom right of the resulted matrix. We are using DCT to generate the feature vector for recognizing a character by selecting the first 64 higher value DCT coefficients that are extracted in a zigzag fashion.

C. Recognition Techniques

Four different systems have been examined in this paper, as follows:

System 1: It uses the GRNN algorithm with an integer output (1-28) that represents a character, i.e. 1 for all Alif patterns, …, 28 for all Yaa patterns. The spread of radial basis functions was 0.2.

System 2: It uses the kNN with an integer output (1-28) that represents a character, i.e. 1 for all Alif patterns, …, 28 for all Yaa patterns. The nearest neighbors number (k) was 1.

System 3: It uses the SVM that relies on 28 SVMs, one rest method. The order of the polynomial kernel was 2.

System 4: It is a combination system of the three previous systems. It works as follow: if at least two of the three members agree, then their results are trusted, otherwise the structural similarity index (SSIM) method is applied between the given pattern and the outputs of the system (1-3). The highest cost was used to identify the recognized character. Consider the systems (1-3) are completely disagree to classify a character. For instance, at noise level equal to 50%, System 1 recognizes the character as Daal ‘د’, System 2 recognizes the character as Raa ‘ر’, and System 3 recognizes the character as Dhaal ‘ذ’. The final decision, therefore, will be the highest SSIM between the given characterand the characters Daal ‘د’, Raa ‘ر’, and Dhaal ‘ذ’.

VI. Results and Discussion

The training set used in this paper is Arabic character images with five 14-point size fonts (Arial, Times New Roman, Tahoma, Courier New, and Adobe Naskh Medium). Table 3 shows examples of these five fonts of each character in the training set. This training set includes 500 patterns. The systems give a full performance on the training dataset. To make testing sets more realistic, three versions of a testing set are generated. Each testing set is the same training set after corrupted by one of the following levels of *“Salt & Pepper”* noise: 10%, 30%, and 50%. Some testing character samples (Arial 14) are illustrated in Table 4.

Table 3.
Samples of font styles used in the study

Font names	Characters
Arial	ا ب ت ث ج ح خ د ذ ر ز س ش ص ض ط ظ ع غ ف ق ك ل م ن ه و ي
Times New Roman	ا ب ت ث ج ح خ د ذ ر ز س ش ص ض ط ظ ع غ ف ق ك ل م ن ه و ي
Tahoma	ا ب ت ث ج ح خ د ذ ر ز س ش ص ض ط ظ ع غ ف ق ك ل م ن ه و ي
Courier New	ا ب ت ث ج ح خ د ذ ر ز س ش ص ض ط ظ ع غ ف ق ك ل م ن ه و ي
Adobe Naskh Medium	ا ب ت ث ج ح خ د ذ ر ز س ش ص ض ط ظ ع غ ف ق ك ل م ن ه و ي

Table 4.
Testing character samples with three levels of noise

Char name	Acquired char	Testing char: Char corrupted with noise level 10%	Testing char: Char corrupted with noise level 30%	Testing char: Char corrupted with noise level 50%
Jeem "ج"				
Daal "د"				
Dhaal "ذ"				
Zaay "ز"				
Waaw "و"				

Figures 2-5 show the results obtained from applying System 1, System 2, System 3 and System 4 for three different noise levels (10%, 30%, and 50%) of the testing set.

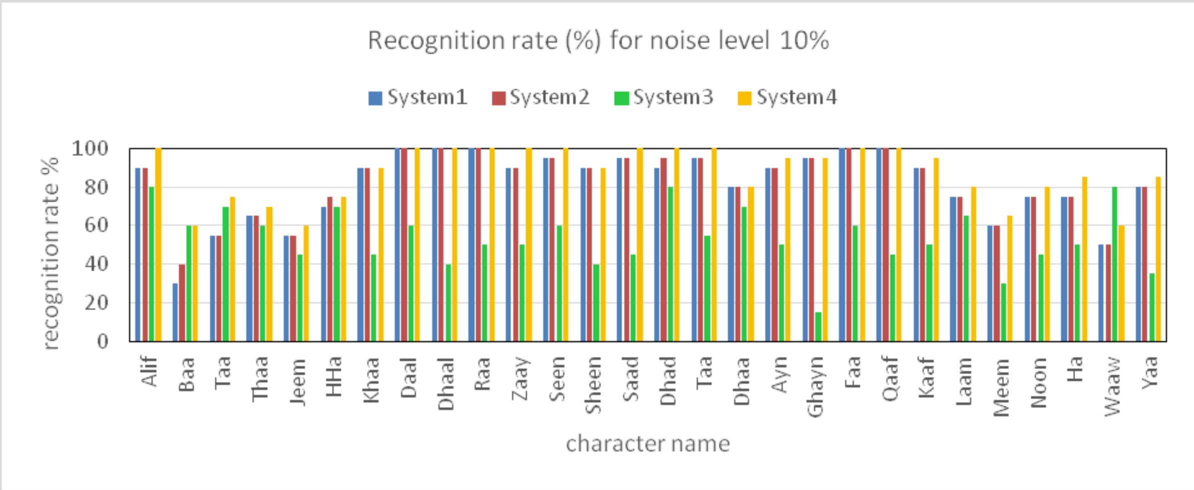

Figure 2. Recognition rate (%) of systems 1-4, noise level 10%

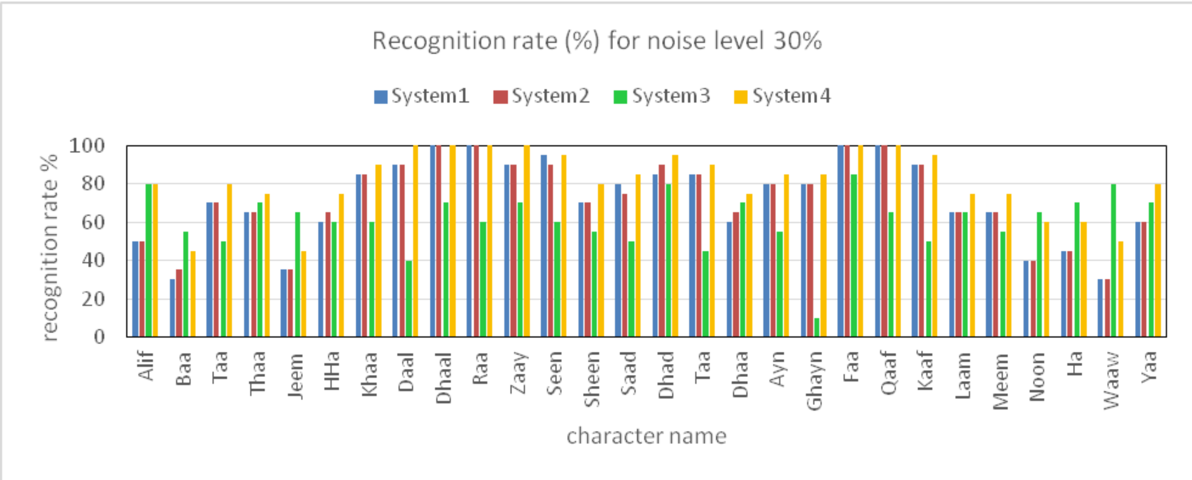

Figure 3. Recognition rate (%) of systems 1-4, noise level 30%

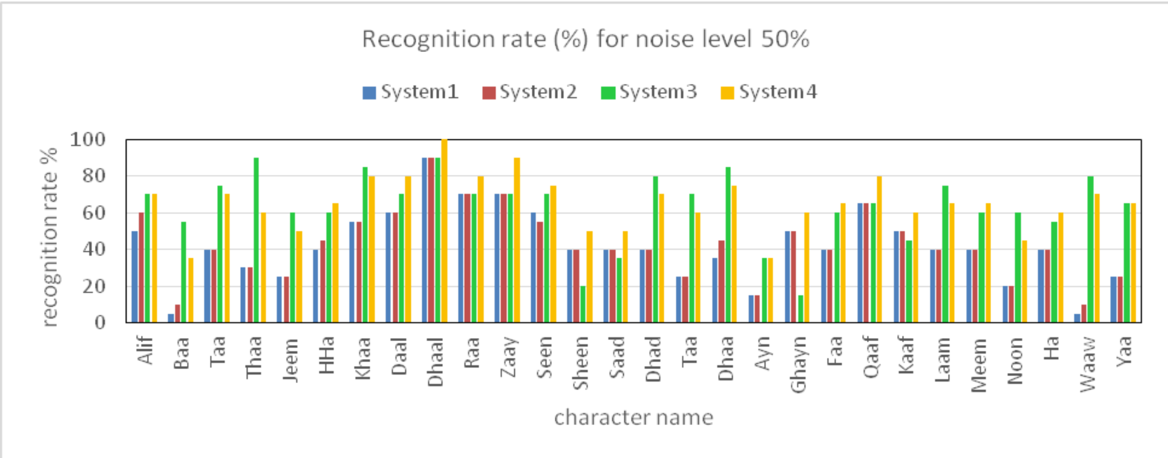

Figure 4. Recognition rate (%) of systems 1-4, noise level 50%

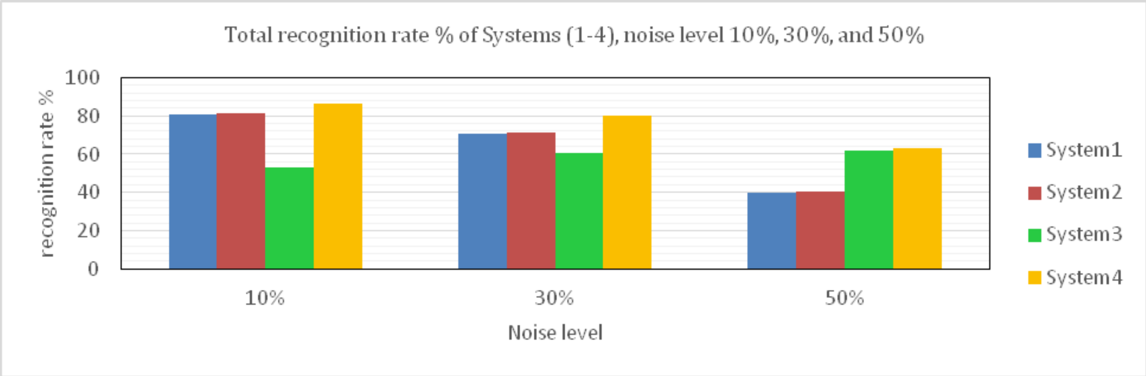

Figure 5. Total recognition rate (%) of systems 1-4 for each noise level

Closer inspection of Figures 2-5 shows that the confidence-based backoff system (System 4) outperform each of the individual systems (System 1, System 2, and System 3) for all three levels of noise. Additionally, a direct and significant association between the performance (recognition rate) and the noise intensity was found. Accordingly, as the noise intensity increased, the performance of systems (1, 2, and 4) decreased, except system 3, a touched increase was observed in its performance.

The results also shows that system 4 gives high performance for the characters (Alif, Khaa, Daal, Dhaal, Raa, Zaay, Faa, and Qaaf), especially for Dhaal character the system gives a full performance for all three levels of noise. Surprisingly, system 3 gives the stable performance for Waaw character for all noise levels and thus outperformed the other systems.

VII. Conclusion

The current work aims at suggesting a technique for Arabic that is able to recognize different fonts of isolated characters. If there are different techniques do the same task but work differently, it seems reasonable to use some wise combination of them to exploit the unique advantage of each one to obtain better performance than obtained by any of them alone. We have investigated strategy for combining three machine-learning techniques for recognition Arabic characters, where imperfect and dimensionally variable input characters are used. The current strategy accepts the outputs of the individual systems if at least two of them agree and backoff to the highest SSIM between the given character and the outputs of the system (1-3) is used to identify the recognized character if they do not. It is supposed that the reason for the effectiveness of the current technique arises from the fact that the individual systems work in fundamentally various ways. Therefore, if the systems make regular mistakes, these will be different. This, in turn, means that the places where the systems work perfectly will be different.

Overall, the current findings were encouraging. The proposed system (System 4) gives better results than each system in isolation (System 1, System 2, and System 3). The total recognition rates of each system were 63.8%, 64.5%, 58.4%, and 76.7% for System 1, System 2, System 3, and System 4 respectively.

References

[1] LORIGO, L.M. and GOVINDARAJU, V. (2006) Offline Arabic handwriting recognition: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28 (5), pp. 712-724.

[2] MAZROUI, A. and KERKOUR ELMIAD, A. (2014) Recognition of Multifont Isolated Arabic Characters by Bézier Curves. *International Journal of Computer Applications*, 100 (6), pp. 1-7.

[3] JADHAV, D. and VEERESH, G. (2012) Multi-Font/Size Character Recognition. *International Journal of Advances in Engineering & Technology*, 3 (2), pp. 442-445.

[4] SAMADIANI, N. and HASSANPOUR, H. (2015) A neural network-based approach for recognizing multi-font printed English characters. *Journal of Electrical Systems and Information Technology*, 2 (2), pp. 207-218.

[5] KAPOOR, R., GUPTA, S., and SHARMA, C. (2011) Multi-font/size character recognition and document scanning. *International Journal of Computer Applications*, 23 (1), pp. 21-24.

[6] BAI, G. (1993) Multifont Chinese character recognition using side-stroke-end feature. In: *Proceedings of the 2nd*

International Conference on Document Analysis and Recognition, Tsukuba Science City, October 1993. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

[7] ZHONG, Z., JIN, L., and FENG, Z. (2015) Multi-font printed Chinese character recognition using multi-pooling convolutional neural network. In: *Proceedings of the 13th International Conference on Document Analysis and Recognition, Tunis, August 2015*. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, pp. 96-100.

[8] AMOR, N.B. and AMARA, N.E.B. (2006) Multifont Arabic Characters Recognition Using Hough Transform and HMM/ANN Classification. *Journal of Multimedia*, 1 (2), pp. 50-54.

[9] KHORSHEED, M.S. (2007) Offline recognition of omnifont Arabic text using the HMM ToolKit (HTK). *Pattern Recognition Letters*, 28 (12), pp. 1563-1571.

[10] KILIC, N., GORGEL, P., UCAN, O.N., and KALA, A. (2008) Multifont Ottoman character recognition using support vector machine. In: *Proceedings of the 3rd International Symposium on Communications, Control and Signal Processing, St Julians, March 2008*. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, pp. 328-333.

[11] IZAKIAN, H., MONADJEMI, S., LADANI, B.T., and ZAMANIFAR, K. (2008) Multi-font Farsi/Arabic isolated character recognition using chain codes. *World Academy of Science, Engineering and Technology*, 43, pp. 67-70.

[12] GIACINTO, G., ROLI, F., and FUMERA, G. (2000) Selection of classifiers based on multiple classifier behaviour. In: FERRI, F.J., IÑESTA, J.M., AMIN, A., and PUDIL, P. (eds.) *Advances in Pattern Recognition. SSPR/SPR 2000. Lecture Notes in Computer Science*, Vol. 1876. Berlin, Heidelberg: Springer, pp. 87-93.

[13] ALABBAS, M. and RAMSAY, A. (2012) Improved POS-Tagging for Arabic by Combining Diverse Taggers. In: ILIADIS, L., MAGLOGIANNIS, I., and PAPADOPOULOS, H. (eds.) *Artificial Intelligence Applications and Innovations. AIAI 2012. IFIP Advances in Information and Communication Technology*, Vol. 381. Berlin, Heidelberg: Springer, pp. 107-116.

[14] ALABBAS, M. and RAMSAY, A. (2014) Improved Parsing for Arabic by Combining Diverse Dependency Parsers. In: VETULANI, Z. and MARIANI, J. (eds.) *Human Language Technology Challenges for Computer Science and Linguistics. LTC 2011. Lecture Notes in Computer Science*, Vol. 8387. Cham: Springer, pp. 43-54.

[15] ALABBAS, M. and RAMSAY, A. (2014) Combining strategies for tagging and parsing Arabic. In: *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Langauge Processing, Doha, October 2014*. Stroudsburg, Pennsylvania: Association for Computational Linguistics, pp. 73-77.

[16] JAF, S., ALABBAS, M., and KHUDEYER, R.S. (2018) Combining Machine Learning Classifiers for the Task of Arabic Characters Recognition. *International Journal of Asian Language Processing*, 28 (1), pp. 1-12.

[17] ALABBAS, M., KHUDEYER, R.S., and JAF, S. (2016) Improved Arabic characters recognition by combining multiple machine learning classifiers. In: *Proceedings of the 2016 International Conference on Asian Language Processing, Tainan, November 2016*. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, pp. 262-265.

[18] SPECHT, D.F. (1991) A general regression neural network. *IEEE Transactions on Neural Networks*, 2 (6), pp. 568-576.

[19] BURKOV, A. (2019) *The Hundred-Page Machine Learning Book*. Andriy Burkov.

[20] AGGARWAL, C.C. (2018) *Neural Networks and Deep Learning*. Springer.

[21] RAO, K.R. and YIP, P. (2014) *Discrete Cosine Transform: Algorithms, Advantages, Applications*. Cambridge, Massachusetts: Academic Press.

[22] HUSSEIN, S.A. and KAREEM, M.R. (2019) A Proposed Arabic Text and Text Image Classification Technique Using a URL Address. *Journal of Southwest Jiaotong University*, 54 (5). Available from

http://jsju.org/index.php/journal/article/view/411.

参考文:

[1] LORIGO，L.M. 和 GOVINDARAJU，V.（2006）离线阿拉伯文手写识别：一项调查。电气工程师学会模式分析与机器智能交易，28（5），第 712-724 页。

[2] MAZROUI，A. 和 KERKOUR ELMIAD，A.（2014）通过贝济耶曲线识别孤立的多字体阿拉伯字符。国际计算机应用杂志，100（6），第 1-7 页。

[3] JADHAV，D. 和 VEERESH，G.（2012）多功能字体/大小字符识别。国际工程技术进步杂志，3（2），第 442-445 页。

[4] SAMADIANI，N. 和 HASSANPOUR，H.（2015）一种基于神经网络的方法，用于识别多字体印刷的英文字符。电气系统与信息技术杂志，2（2），第 207-218 页。

[5] R. KAPOOR，S。GUPTA 和 C. SHARMA。（2011）多字体/大小字符识别和文档扫描。国际计算机应用杂志，23（1），第 21-24 页。

[6] BAI，G.（1993）使用侧击结束特征的多字体汉字识别。于：1993 年 10 月在筑波科学城举行的第二届国际文档分析与识别国际会议论文集。新泽西州皮斯卡塔维：电气与电子工程师学会。

[7] 钟钟，金林 和 冯征（2015）使用多池卷积神经网络的多字体印刷汉字识别。于：2015 年 8 月在突尼斯举行的第 13 届文档分析和识别国际会议论文集。新泽西州皮斯卡塔维：电气与电子工程师协会，第 96-100 页。

[8] AMOR，N.B。和 NARA 的 AMARA（2006）使用踝关节变换和 HMM /人工神经网络分类的多字体阿拉伯字符识别。多媒体杂志，1（2），第 50-54 页。

[9] M.S. KHORSHEED（2007）使用 HMM 工具包（HTK）脱机识别多功能字体阿拉伯文本。模式识别字母，28（12），第 1563-1571 页。

[10] KILIC，N.，GORGEL，P.，UCAN，O.N. 和 KALA，A.（2008）使用支持向量机的多字体奥斯曼字符识别。在：第三届通信，控制和信号处理国际研讨会论文集，圣朱利安斯，2008 年 3 月。新泽西州皮斯卡塔维，电气和电子工程师协会，第 328-333 页。

[11] IZAKIAN，H.，MONADJEMI，S.，LADANI，B.T. 和 ZAMANIFAR，K.（2008）使用链码的多字体波斯/阿拉伯语隔离字符识别。世界科学，工程和技术研究院，43，第 67-70 页。

[12] GIACINTO，G.，ROLI，F. 和 FUMERA，G.（2000）基于多个分类器行为的分类器选择。在：FERRI，F.J.，IESTA，J.M.，AMIN，A。和 PUDIL，P。（编辑）模式识别方面的进展。SSPR / SPR2000。计算机科学讲义，第 1 卷。1876 年。柏林，海德堡：施普林格，第 87-93 页。

[13] ALABBAS，M. 和 RAMSAY，A.（2012）通过结合不同的匕首改进了阿拉伯语的销售点标签。在：ILIADIS，L.，MAGLOGIANNIS，I. 和 PAPADOPOULOS，H.（编）人工智能应用与创新。联合会 2012。《联合会信息和通信技术的进步》，第 1 卷。 381.柏林，海德堡：施普林格，第 107-116 页。

[14] ALABBAS，M. 和 RAMSAY，A.（2014）通过组合多样的依赖性解析器来改进阿拉伯语的解析。于：VETULANI，Z. 和 MARIANI，J.（编.）对计算机科学和语言学的人类语言技术挑战。LTC2011。计算机科学讲义，第 1 卷。8387. 湛：施普林格，第 43-54 页。

[15] ALABBAS，M. 和 RAMSAY，A.（2014）标记和解析阿拉伯语的组合策略。于：2014 年 10 月，多哈，EMNLP 2014 年阿拉伯自然语言处理研讨会讲稿，宾夕法尼亚州斯特劳兹堡：计算语言学协会，第 73-77 页。

[16] S. JAF，M。ALABBAS 和 R.S. KHUDEYER。（2018）结合用于阿拉伯字符识别任务的机器学习分类器。国际亚洲语言处理杂志，28（1），第 1-12 页。

[17] ALABBAS，M.，KHUDEYER，R.S. 和 JAF，S.（2016）通过组合多个机器学习分类器改进了阿拉伯字符识别。于：2016 年亚洲语言处理国际会议论文集，台南，2016 年 11 月。新泽西州皮斯卡塔维：电气与电子工程师协会，第 262-265 页。
[18] SPECHT，博士（1991）通用回归神经网络。电气工程师学会神经网络交易，2（6），第 568-576 页。
[19] BURKOV，A。（2019）一百页机器学习书。安德烈·伯科夫（Andriy Burkov）。
[20] AGGARWAL，抄底（2018）神经网络与深度学习。施普林格。
[21] RAO，K.R。和 YIP，P.（2014）离散余弦变换：算法，优势，应用。马萨诸塞州剑桥：学术出版社。
[22] HUSSEIN，S.A. 和 KAREEM，M.R.（2019）一种使用网址地址的拟议阿拉伯文本和文本图像分类技术。西南交通大学学报，54（5）。可从 http://jsju.org/index.php/journal/article/view/411 获得。