Kuwait J. Sci. **47 (3)** pp. 2-13, 2020

Deep learning autoencoder approach: Automatic recognition of artistic Arabic calligraphy types

Rami Al-Hmouz

Department of Electrical and Computer Engineering,
King Abdulaziz University, Jeddah, Saudi Arabia
corresponding author: ralhmouz@kau.edu.sa

Abstract

Recognition of Arabic calligraphy types is a challenging problem. Difficulties include similarities among different types, overlap between letters, and letters that assume different shapes. In this study, a deep learning approach to recognizing artistic Arabic calligraphy types is presented. Autoencoder is a deep learning approach with the capability of reducing data dimensions in addition to extract features. Autoencoders can be stacked with several layers. The system is composed of three layers consisting of two encoder layers to extract features and one soft max layer for the recognition stage. The font can be recognized in a collective manner based on the words or segments that exist in the font images. The input of the system consists of individual words or segment images that compose the font image, and the output is the recognized font type. The approach was evaluated on local and public datasets, and the achieved recognition rates were 92.1% and 99.5%, respectively.

Keywords: Artistic Arabic calligraphy; autoencoder; deep learning; optical font recognition.

1. Introduction

Arabic calligraphy is the artistic writing for languages such as Arabic, Urdu, and Persian, that use Arabic letters. Calligraphy is part of the heritage of Islamic and Arabic cultures (Weitzel, 2005). There are several types of Arabic calligraphy, such as Thuluth, Naskh, Dewani, Reqa'a, Jali Dewani, Farsi, and Kufi (Figure 1). The most commonly used style is Reqa'a. Calligraphers can identify the calligraphy type based on the geometric shapes, structure, and overlapped words and letters. Arabic text is written from right to left from an alphabetic set containing 28 letters. The letters have only 18 different shapes. These shapes might come with dots (diacritics) on the top or bottom of letters, or they may be contained inside to form distinctive letters.

There are many applications that are related to Arabic language, such as steganography (Gutub *et al.*, 2010), cryptography (Avcı, 2016), natural language processing (Guellil, 2019), and detection and recognition of text (Yousfi, 2016; Khan *et al*., 2018) . Optical character recognition (OCR) is the process of identifying characters from images, while optical font recognition (OFR) is the process identifying the font type from text images (Zramdini and Rolf, 1998). These text images could be drawn by professional calligraphers or generated by computer applications. Arabic font recognition is a complex problem because Arabic words are formed from characters that have to be connected in different ways, and words might be tangled together or even curved in order to form the required font type. Moreover, some punctuation symbols could appear below, above, or along with the text according to the calligrapher's style, which might make the text look like drawings as shown in Figure 2.

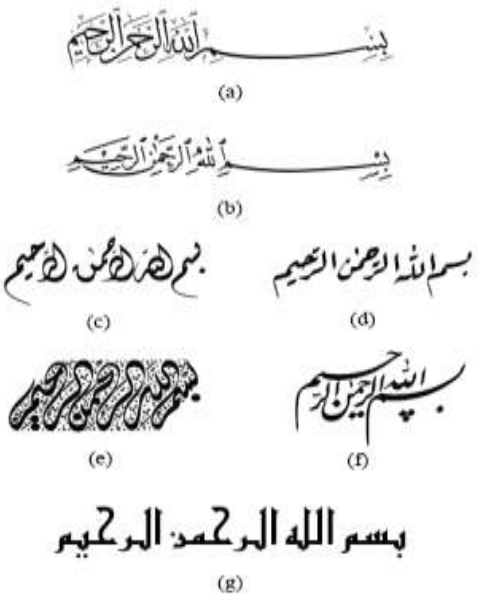


Fig. 1. Samples of different artistic Arabic calligraphy types: (a) Thuluth, (b) Naskh, (c) Dewani, (d) Reqa'a, (e) Jali Dewani (f) Farsi (g) Kufi

Fig. 2. Samples of artistic Arabic calligraphy

Like any pattern recognition problem, features are extracted from text images and then are fed into a classifier. Feature extraction might be achieved locally at the word level or globally at the image level (Zhu *et al.*, 2001).

Researchers have been investigating global feature approaches for Arabic fonts. Statistical analyses on features have been carried out on Arabic font images that include edges (Bilal *et al.*, 2011), morphological features (Allaf and Al-Hmouz, 2016), a gray level co-occurrence matrix (Bilal *et al.*, 2012), and horizontal and vertical projection profiles (Ibrahim, 2005). Other different feature extraction methods for Arabic font images have been studied. These include fractal geometry (Moussa *et al.*, 2006; Moussa *et al.*, 2010), wavelets (Zaghden *et al.*, 2006), and Sobel–Roberts features (Hossein and Ehsanollah, 2010). Furthermore, in regards to local methods, recognition systems of types, formats, and sizes of Arabic font images were investigated by Slimane *et al*. (2013). They used a stochastic approach on a sliding window of 102 features. Lutf *et al.* (2014) proposed the segmentation of diacritics of images in order to extract their composite of central and ring projection features.

Autoencoders are one of the key functional models of deep learning architecture. Data representation, compression, abstraction, and feature extraction arise in the field of big data. An autoencoder generates a reduced and representative feature space that is the key point of effective processing in classification or regression problems (Salakhutdinov and Hinton, 2006; Deng *et al.*, 2014). Features can be obtained from the output of the hidden layer and then are propagated to the next layers with more reduction in the feature space in the cascaded architecture of autoencoders.

There have been several studies that have focused on various architectural refinements of autoencoders (Gao *et al*., 2015) (Xia *et al*, 2016), their optimization processes (Dizaji *et al.*, 2017) (Deng *et al.*, 2013) (Chen *et al.*, 2018), and application studies (Janowczyk *et al.*, 2017), (Grozdić *et al.*, 2017). Autoencoders can be generally categorized into denoising, contractive, variational, and k-sparse autoencoders (Vincent *et al*., 2010; Rifai *et al*., 2011; Wu *et al.*, 2019; Ranzato *et al.*, 2007).

Arabic text images can appear in different font types. Therefore, the improvements of OFR will be directly reflected on OCR. OFR could be useful in the learning process of calligraphers to measure the quality of their calligraphy types. Furthermore, in logical documents, the logical label could be defined based on font types for chapter/section titles. This would be a helpful way to recognize the structure of the documents.

Most Arabic calligraphy recognition approaches focus on features at a global level. However, several challenges are presented when considering this approach: (1) number of existing words in text images; (2) words look alike in several font types; and (3) images have different sizes in terms of font and resolution. In contrast, approaches at the local level fail to extract words imaged in the absence of spaces between words in the text images.

A cascaded autoencoder along with a classification layer is presented in this paper. The text image is segmented into smaller images, and the recognition result of the segmented images is fused to reach the final recognition. This paper is important because it describes:

(1) The introduction of a deep learning approach in the Arabic calligraphy recognition problem: deep learning architectures have been proven to achieve competitive accuracy compared to other methods in many applications.

(2) The segmentation of font images: the font image is segmented to several images to capture the local

characteristics of font types in the recognition process.

(3) The fusing the classifier outputs of the segmented images: the scores of the recognized segmented images are quantified in a probability score, which is the key aspect of improving the performance of the deep learning structure.

The paper is structured into five sections. Section 2 outlines the main ideas of autoencoders along with the related design problems. Section 3 is concerned with the Arabic font type recognition approach in which the idea of fusing recognition results is incorporated into the topology of the system. Section 4 is devoted to experimental studies. Concluding comments are covered in Section 5.

1. Autoencoder architecture and design

Before proceeding with the introduction of the Arabic calligraphy recognition approach, a brief description of the autoencoder concept and design is presented. The concept of the autoencoder structure (Deng *et al.*, 2014) is to compress the data of a given vector $\boldsymbol{x}$ (n-dimensional space) into a representative form of the hidden layer of m processing unit ($m<<n$). The data conversion part from n to m dimensional spaces is called the encoder, while converting the hidden layer of m to n back to dimensional spaces is called the decoder. Generally, the encoding and decoding mechanisms are realized in terms of the following nonlinear transformations:

Encoding

$$z = f(Wx + b)$$

$$z_j = f(\sum_{i=1}^{n} w_{ij}x_i + b_j) \quad j=1, 2, \ldots, m \quad , \tag{1}$$

in which $\boldsymbol{W}$ is the weight matrix of dimensionality n by m, $\boldsymbol{b}$ stands for a vector of bias (m dimensional space), and f is a certain nonlinear mapping (typically a monotonically increasing function (say a sigmoidal one).

Decoding

Decoding produces the output vector computed in the following form:

$$\hat{x} = f(Vz + g)$$

$$\hat{x}_i = f(\sum_{j=1}^{m} v_{ij}z_j + g_i) \quad i=1, 2, \ldots, n \quad , \tag{2}$$

in which $\boldsymbol{V}$ is the weight matrix of the decoder (mXn), and $\boldsymbol{g}$ is the vector of biases of the output processing units (n dimensional space). The overall processing realized by the auto-encoder is shown in Figure 3.

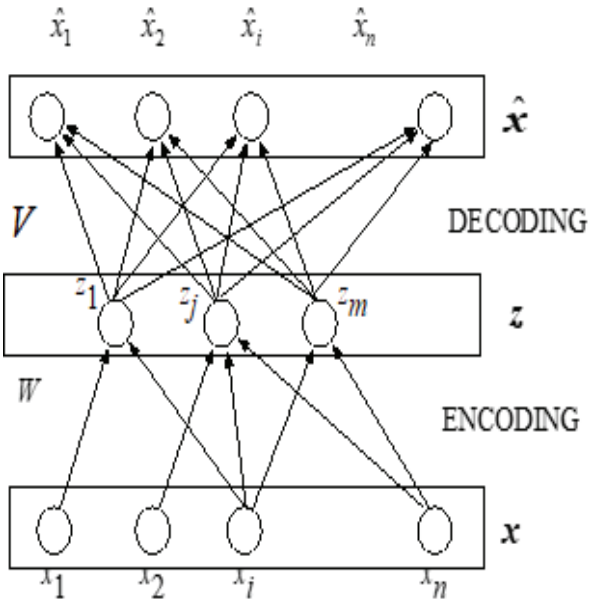


Fig. 3. Architecture of auto-encoder

The goal of an autoencoder is to transform the data ($\boldsymbol{x}$) into an abstract compressed version in the encoder part and transfer it back again with some error ($\hat{\boldsymbol{x}}$) in the decoder part. The compression can be measured by the ratio m/n in which high compression is produced by a low ratio. The design of the autoencoder is a minimization process of (3) in order to find the optimal values of the matrices, $\boldsymbol{W}$,$\boldsymbol{V}$, and the vectors, $\boldsymbol{b}$ and $\boldsymbol{g}$.

$$Q = \frac{1}{n}\sum_{k=1}^{n} \|x(k) - \hat{x}(k)\| \tag{3}$$

Autoencoders can be stacked one over another as illustrated in Figure 4. The shaded layers (compressed data) represent the successively produced abstract features (z)

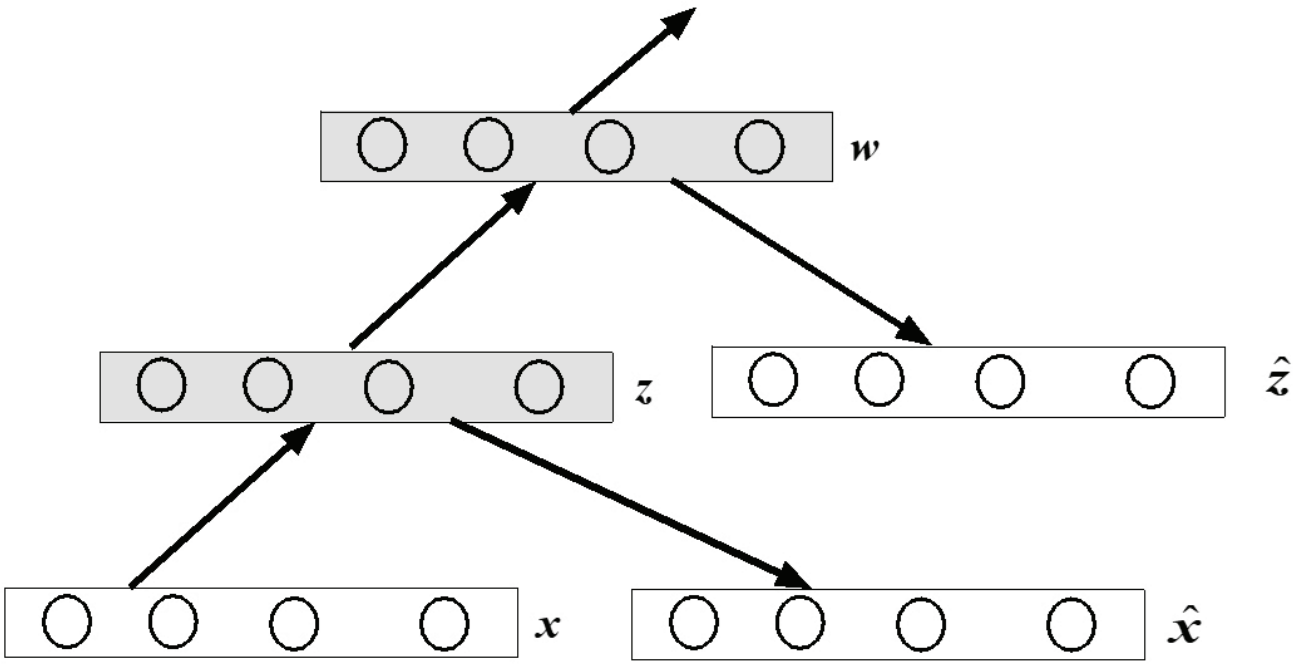


Fig.4 . Stacked autoencoders with two-layered architecture

2. Autoencoder-based automatic recognition of artistic Arabic calligraphy types

As mentioned earlier, an autoencoder is used to capture representative features. Hence, a new layer could be added to the autoencoder structure in order to achieve the recognition process. A soft max layer, which assigns probabilities to each font type, is connected at the last stage. The sum of those probabilities is 1.0. The soft max layer is considered the output layer in which the whole structure of the network is trained in a supervised process (Figure 5).

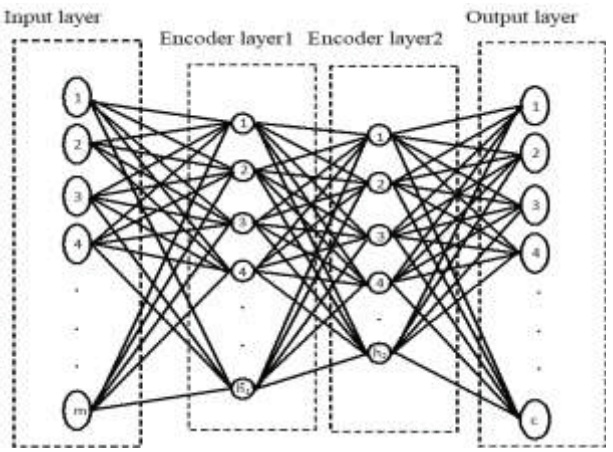


Fig. 5. Arabic calligraphy font type recognition architecture

The raw data from the text images are fed to the input layer, and the output layer produces the recognized font type. Text images come with different sizes according to the number of words present in the image. Words can also be written in different font sizes according to the selected pen. Normalizing the input images to a fixed size is a prerequisite in the process of deep learning algorithms. This normalization will compress/ enlarge the input images and degrade their quality in terms of distinctive features among all types. This process could lead to an increase in the error rate. Accordingly, the text image could be segmented into the number of images in relation to the number of words existing in the text. This segmentation can easily be achieved if there are clear spaces between words. However, some words in some Arabic calligraphy types are tangled in such a way that segmentation of the word images is almost an impossible task. In this case, the image is segmented into five portions of equal sizes (Figure 6). It should be noted that segments should be normalized to a fixed size in order to meet the requirements of the deep learning structure. We assumed word-sized segments could reduce the effects of image resizing compared to resizing of text images.

Fig 6. Samples of segmented Arabic text images

For each input (segment) in the deep learning structure, there will be an output (recognized font type) associated with it. Segment outputs that are associated with the same text image are combined by multiplying their scores (refer to 4). A decision of the recognized font type is made according to the highest score through the normalization process in (5). This process is achieved during the training phase in order to compute the error in each iteration in the process of back propagation. The error is computed for the original text image instead of individual segmented images. The features of font type are imbedded in the segmented images, and those segments that do not have enough features for a specific font type will be eliminated in the fusion process at the output layer.

$$o_i = \prod_{k=1}^{\text{no of segment}} s_i, \quad i=1,2,\ldots c \qquad (4)$$

normalization of o:

$$O_i = \frac{o_i}{\sum_{j=1}^{c} o_j} \qquad i=1,2,\ldots c \qquad (5)$$

The process of recognized font type is depicted in Figure 7. The fusion results of the classification scores-obtained by individual segment-are crucial for improving the classification rate. For example, the individual classification scores of the segmented images for a three-class problem are:

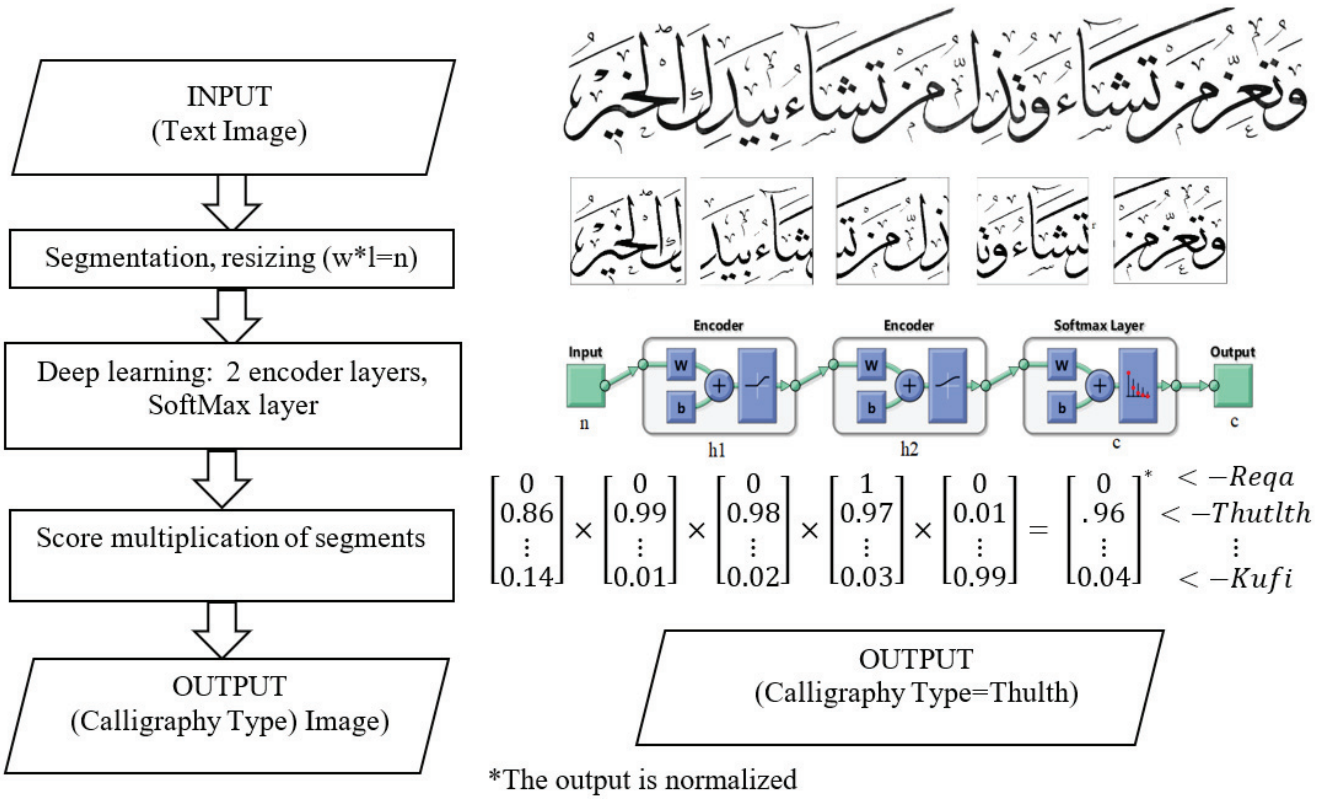


Fig. 7. Flowchart and graphic illustration of autoencoder-based Arabic calligraphy font type recognition method

Although segment 5 has a low probability (score) to class 2, the result after the fusion is still class 2 because it has high probabilities with respect to the other segments. The fused scores can be also normalized by:

$$\frac{\begin{bmatrix} 0 \\ 0.829 * 10^{-3} \\ 2.44 * 10^{-9} \end{bmatrix}}{(0 + 0.829 * 10^{-3} + 2.44 * 10^{-9})} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

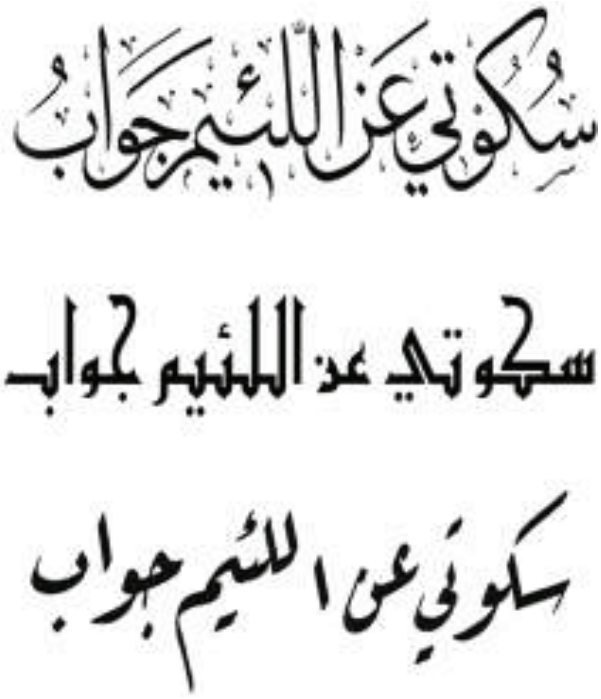
Therefore, the decision of the system will be class 2. Generally, misclassification could arise in some segments, and those segments may have probability values for the incorrect class. Other segments could support the correct class and most probably would have a zero or small values for the incorrect class. If the incorrect class is supported by one or two segments, it will be eliminated in the other segments, especially when they have zero probability. The segmentation process helps in capturing the properties of the font type in at least one segment in addition to reducing the error rate.

3. Experimental studies

This section evaluates the proposed method for Arabic font types on images that have been generated by computer applications (public dataset) and for images that have been written by a professional calligrapher (local dataset).

Local data set: Full sentences and individual words were written by a professional calligrapher, each in three fonts (Thuluth, Reqaa, and Kufi). The texts were scanned and saved in image formats. The images had different characters in terms of number of words, resolution, and font size. In total, there were 131 texts of Arabic calligraphy images. We segmented only text images into five individual images (words images remain untouched). The total number of images after segmentation was 421 images. Then the segmented images were randomly combined in order to form a text. The meaning of the Arabic text in the dataset was considered irrelevant to this study. The dataset was then randomly split into three sets: (1) 35% training; (2) 35% validation, and (3) 30% testing. Samples from the local set are displayed in Figure 8a.

Public data set (APTI): Computer-based fonts were generated (https://diuf.unifr.ch/diva/ APTI/). In this set, there were 10 font types of different sizes and styles (normal, italic, and bold). Only word images were available. For each font type, there were 10 sizes and four font styles. We tested our method on data that had a font size of 14 and was in bold. In the selected set, there were 944,180 images. This number was normally distributed among all font types. The data were randomly grouped into three sets: (1) 2.5% training; (2) 2.5% validation, and 95% testing. Samples of the public set are shown in Fig. 8b.



(a)

(b)

Fig. 8. Script samples: (a) local set (b) public set

The sets included either segmented images or already saved word images. In order to form text images, word images were combined at random into two, three, four, five, and random combinations of the same font type (refer to Table 1). This ensured that results were not biased to certain combinations of images in the sets. The process of recognition was repeated 10 times. During each iteration, the datasets were formed randomly, and several cases were tested to represent the number of words that compose the font image (refer to Table 2). The average results of the iterations were reported as the final recognition rates. The function and other variables for the system are shown in Table 3.

Table 1. Summary of local/public data sets used in experiment

Set	Calligraphy Type	Training	Testing
Local	Thuluth	96	41
	Reqaa	109	47
	Kufi	90	38
APTI	Advertising Bold	4721	89697
	Andalus	4721	89697
	Arabic Transparent	4721	89697
	DecoType Naskh	4721	89697
	DecoType Thuluth	4721	89697
	Diwani Letter	4721	89697
	MUnicode Sara	4721	89697
	Simplified Arabic	4721	89697
	Tahoma	4721	89697
	Traditional Arabic	4721	89697

In the public set, the number of nodes at the input layer was selected from the average size of the images in the training set (23*58), and the image was then arranged in a vector form of 1334. The images in the local set were randomly resized to (92*232) because most of the scanned Arabic calligraphy images have high resolution (>1000*1000), which makes the training process a difficult task using a processors with a normal speed (several GHz). Hence, the number of nodes in the input layer was set to 21,344. In regards to the number of nodes in the first autoencoder, 1,000 images were tested in order to evaluate the error rate for different values of the hidden layer (h_1) as shown in Figure 9. The knee point was marked at h=100 for the first encoder, and for h_2, the value was set to 50.

Table 2. Summary of local/public data sets used in experiment

Set		APTI	Local
2 words	Training	23610	148
	Testing	448490	6
3 words	Training	15740	99
	Testing	298900	43
4 words	Training	11810	75
	Testing	224250	33
5 words	Training	9450	50
	Testing	179400	22
random words (2,3,4,5)	Training	10507	67
	Testing	199345	29

Table 3. Summary of functions and variables used in the deep learning architecture

Set	Input layer	Autoencoder 1	Autoencoder 2	Soft max
APTI	No of nodes 23X58 (1334)	.h1 =100 .Encoder Function: satlin .Decoder Function: purelin .Max Epochs= 1000	.h2 =50 .Encoder Function: satlin .Decoder Function: purelin .Max Epochs= 1000	.No of nodes= 10 .Training Algorithm: trainscg* .Loss Function: cross entropy .Max Epochs= 1000
Local	No of nodes 92X232	.h1 =100 .Encoder Function: logsig .Decoder Function: logsig .Max Epochs= 1000	.h2 =50 .Encoder Function: logsig .Decoder Function: logsig .Max Epochs= 1000	.No of nodes=3 .Training Algorithm: trainscg* .Loss Function: cross entropy .Max Epochs= 1000

* scaled conjugate gradient

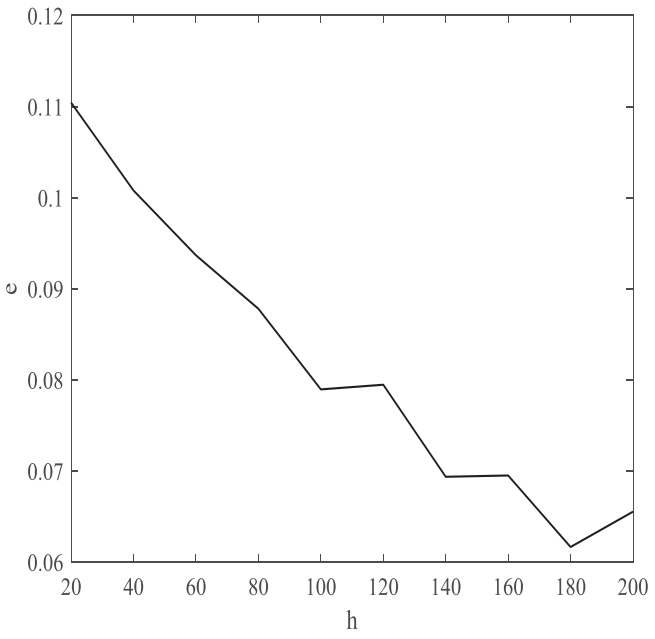


Fig. 9. Mean square error versus number of nodes of first autoencoder

Table 4 shows the recognition rates for the local set using a different number of segments. For an example, three segments are referred to as all font images are split into three images of equal size, and in the last case (random 2,3,4,5), images were split randomly in up to 5 segments. The recognition rate clearly increased when image segmentation of the text images was used. The best achievement occurred when four segments were used. As mentioned earlier, the segmentation was blind, which means that each segment did not necessary represent a word image. Some segments in the local set represent only a character or even part of a character. The improvement in segmentation (4 segments) over no segmentation of font images achieved 25%.

Table 4. Recognition rate of local set

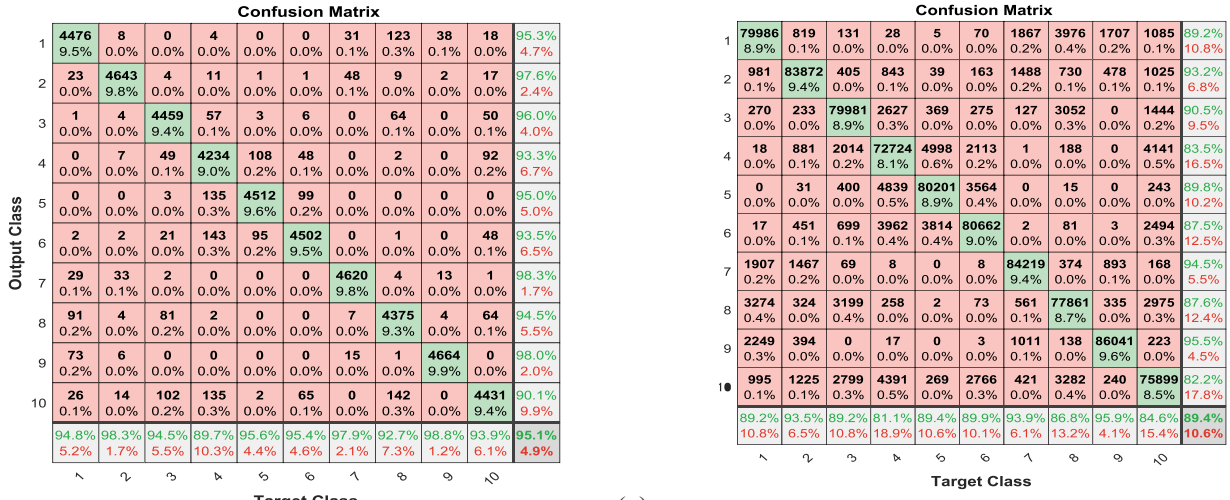
No. of segments	Training	Testing
-	98.59	73.20
1	99.0	79.1
2	99.3	85.8
3	99.8	90.0
4	100.00	92.4
5	100.00	92.2
Random (2,3,4,5)	99.8	92.1

The recognition rates for the public set are given in Table 5. As expected, the recognition rate increased as the number of segments increased. More segments led to better recognition rates. Each segment represented a word image. Therefore, including more segments improved the accuracy of font recognition. Figure 10 shows results for the confusion matrix for the training and test sets when using one segment and random number of segments in representing the text image.

Table 5. Recognition rate of public set APTI

No. of segments	Training	Testing
1	95.1	89.4
2	99.12	97.1
3	99.8	99.0
4	99.8	99.6
5	99.9	99.8
6	100	99.9
Random (2,3,4,5)	99.9	99.5

Table 6 compares the proposed method and methods reported in the literature regarding recognition rate. It should be noted that the experimental setup was different in the selected data sets. Differences included training/testing ratio, validation procedure, and image size. study In the study by Slimane *et al.* (2013), which examined the APTI dataset, the ratio of was (1/1). In our experiment, a very low ratio of 1/19 (5% training, 95% testing) was employed in order to show that a high recognition rate could be achieved using the image segmentation method with the deep learning approach. Although a high recognition rate using only 5% of the data in the training phase was achieved, increasing this ratio to (1/1) in our study definitely increases the recognition rate. The recognition rate also increased once more segments were involved in the process.



(a)

(b)

Fig. 10. Confusion matrices for training and testing sets of public set (APTI) for number of segments used in recognition: (a) one segment (b) random segments

Table 6. Reported recognition rates of literature methods

Method		Set Size	No. fonts	(APTI) (%)	Other set (%)
(Allaf & Al-Hmouz, 2016)		944,180	10	91.98	-
(Ibrahim, 2005)		108000	3	-	90.8
(Ben Moussa *et al*., 2006)		450	10	-	98.0
(Ben Moussa *et al*., 2010)		1000	10	-	96.6
(Zaghden *et al*., 2006)		2500	10	-	96.5
(Bilal *et al*., 2012)		700	7	-	98.0
(Hossein & Ehsanollah, 2010)		20000	10		94.2
(Slimane *et al*., 2013)		200000	10	94.5	-
Proposed method	1 segment	944,180	10	89.4	-
	(2,3,4,5) segments	209,852	10	99.5	-

4. Conclusions

Two cascaded layers of autoencoders and a classification layer were presented with the aim of recognizing Arabic font type images. Image segmentation provided local properties of font images, although the deep learning structure provided the capability of producing a high recognition rate. Combining segment results ensured a better recognition rate. The achieved recognition rate in the APTI was 99.5%, while in the local set, it was 92.4%. Improving image segmentation will increase recognition rates, especially for Arabic calligraphy images. For future studies, CNN could be utilized in recognition of font types because it has been proven its ability in many complex applications. There are also generative models like the variational autoencoder and generative adversarial network that could be investigated to generate artistic Arabic calligraphy images.

Acknowledgments

This project was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, Saudi Arabia under grant no. (G/362/135/37). The authors, therefore, acknowledge with thanks DSR technical and financial support.

References

Allaf, S. & Al-Hmouz, R. (2016). Automatic recognition of artistic Arabic calligraphy. JKAU: Eng., Sci, **27**(1): 3-17.

Avcı, D. (2016). A novel meaningful secret image sharing method based on Arabic letters. Kuwait Journal of Science & Engineering, **43**(4): 114-124.

Ben Moussa, S. Zahour, S., Kherallah, A. Benabdelhafid, M. & Alimi, A. (2006). Utilisation de nouveaux paramètres à base de fractale pour la discrimination des fontes arabes. In: Likforman-Sulem, L. (Ed.), Actes du 9ème Colloque International Francophone sur l'Ecrit et le Document, 283-288.

Ben Moussa, S., Zahour, A., Benabdelhafid, A. & Alimi, A.M. (2010). New features using fractal multi-dimensions for generalized Arabic font recognition. Pattern Recognition Letters, **31**(5): 361-371.

Bilal, B., Abdullah, S.N. & Omar, K. (2011). A statistical global feature extraction method for optical font recognition. Intelligent Information and Database Systems, ACIIDS: Asian Conference on Intelligent Information and Database Systems, 3rd International Conference, Korea, 20-22 April. Proceedings, Part I. N.T. Nguyen, C.-G. Kim, & A. Janiak (Eds.): 257-267.

Bilal, B., Abdullah, S.N. & Omar, K. (2012). A novel statistical feature extraction method for textual images: Optical font recognition. Expert Systems with Applications, **39**(5): 5470-5477.

Chen, Z., Yeo, C.K., Lee, B.-S., Lau, C.T. & Jin, Y. (2018). Evolutionary multi-objective optimization-based ensemble autoencoders for image outlier detection. Neurocomputing, **309**: 192-200.

Deng, J., Zhang, Z. Marchi, E. & Schuller, B. (2013). Sparse autoencoder based feature transfer learning for speech emotion recognition. 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction (ACII), 511-516.

Deng, J., Zhang, Z., Eyben, F. & Schuller, B. (2014). Autoencoder-based unsupervised domain adaptation for speech emotion recognition. IEEE Signal Process Letters, **21(9)**: 1068-1072.

Dizaji, K.G., Herandi, A., Deng, C., Cai, W. & Huang, H. (2017). Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In IEEE Int. Conf. Comput. Vis. (ICCV), 5747-5756.

Gao, S. Zhang, Y. Jia, K. Lu, J. Zhang, Y. (2015). Single sample face recognition via learning deep supervised autoencoders. IEEE Transactions on Information Forensics and Security, **10**(10): 2108-2118.

Grozdić, Đ.T., Jovičić, S.T. & Subotić, M. (2017). Whispered speech recognition using deep denoising autoencoder. Engineering Applications of Artificial Intelligence, **59**: 15-22.

Gutub, A., Ghouti, L., Elarian, Y.S., Awaideh, S. Alvi, A. (2010). Utilizing diacritic marks for Arabic text. Kuwait Journal of Science & Engineering, **37**(1): 89-109.

Guellil, I., Saadane, H., Azouaou, F. Gueni, B. & Nouvel, D. (2019). Arabic natural language processing: An overview. Journal of King Saud University - Computer and Information Sciences, ISSN 1319-1578,

https://doi.org/10.1016/j.jksuci.2019.02.006..

Hossein, K. & Ehsanollah, K. (2101). Farsi font recognition based on Sobel–Roberts features. Pattern Recognition Letters, **31**(1): 75-82.

Ibrahim, A. (2005). Arabic font recognition using decision trees built from common words. J. Computing and Information Technology, **13**(3): 211-222.

Janowczyk, A., Basavanhally, A. & Madabhushi, A. (2017). Stain normalization using sparse autoEncoders (StaNoSA): Application to digital pathology. Computerized Medical Imaging and Graphics, **57**: 50-61.

Khan, S., Ali, H., Ullah, Z., Minallah, N. Maqsood, S. & Hafeez, A. (2018). KNN and ANN-based recognition of handwritten Pashto letters using zoning features. International Journal of Advanced Computer Science and Applications, **9**(10): 570-577.

Lutf, M., Xinge, Y., Cheung, Y-M. & Philip Chen, C.L. (2014). Arabic font recognition based on diacritics features. Pattern Recognition, **47**(2): 72-684.

Ranzato, M.A., Poultney, C., Chopra, S. & Cun, Y.L. (2007). Efficient learning of sparse representations with an energy-based model. In Advances in Neural Information Processing Systems 19-Proceedings of the 2006 Conference, Canada, 47 December, 1137-1144.

Rifai, S., Vincent, P., Muller, X., Glorot, X. & Bengio, Y. (2011). Contractive auto-encoders: Explicit invariance during feature extraction. In Proceedings of the 28th International Conference on Machine Learning, Washington, USA, 833-840.

Salakhutdinov, R. & Hinton, G. (2012). An efficient learning procedure for deep Boltzmann machines. Neural Computing, **24**(8): 1967-2006.

Slimane, F., Kanoun, S., Hennebert, J. Alimi, A.M. & Ingold, R. (2013). A study on font-family and font-size recognition applied to Arabic word images at ultra-low resolution. Pattern Recognition Letters, **34**: 209-218.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y. & Manzagol, P.-A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning, **11**(12): 3371-3408.

Weitzel, C. (2005). The written word in Islamic art. In Cusack, C.M. Di Lauro, F. & Hartney, C. (Eds), The Buddha of Suburbia: Proceedings of the Eighth Australian and International Religion. Literature and the Arts Conference 2005. RLA Press, Sydney, Australia, 213-222.

Wu, C., Wu, F., Wu, S., Yuan, Z., Liu, J. & Huang, Y. (2019). Semi-supervised dimensional sentiment analysis with variational autoencoder. Knowledge-Based Systems, **165**: 30-39.

Xia, C., Qi, F. & Shi, G. (2016). Bottom–up visual saliency estimation with deep autoencoder-based sparse reconstruction. IEEE Transactions on Neural Networks and Learning Systems, **27**(6): 1227-1240.

Yousfi, S. (2016). Embedded Arabic text detection and recognition in videos, PhD thesis. Document and Text Processing. Université de Lyon. English. (NNT : 2016LYSEI069).

Zaghden, N., Ben Moussa, S., Alimi, S. & Adel, M. (2006). Reconnaissance des fontes arabes par l'utilisation des dimensions fractales et des ondelettes. In: Likforman-Sulem, L. (Ed.), Actes du 9ème Colloque International Francophone sur l'Ecrit et le Document, 277-282.

Zramdini, A. & Rolf, I. (1998). Optical Font Recognition Using Typographical Features. IEEE Transaction on Pattern Analysis and Machine Intelligence, **20**(4): 877-882.

Zhu, Y., Tan, T. & Wang, Y. (2001). Font recognition based on global texture analysis. IEEE Transaction on Pattern Analysis and Machine Intelligence, **23**(10): 1192-1200.

Submitted: 17/09/2019
Revised: 25/11/2019
Accepted: 01/12/2019

منهج التعلم التلقائي العميق: التمييز التلقائي لأنواع فن الخط العربي

رامي الحموز
قسم الهندسة الكهربائية وهندسة الحاسبات، جامعة الملك عبد العزيز،المملكة العربية السعودية
ralhmouz@kau.edu.sa

ملخص

يشير هذا البحث الي أن التعرف على أنواع فن الخط العربي يمثل إشكالية كبيرة. تشمل هذه الصعوبات وجود تشابهات بين الأنماط المختلفة والتشابك بين الحروف وكذلك الحروف التي تتخذ أشكالا مختلفة. يقدم هذا البحث عرضًا لمنهج التعلم العميق الذي يساعد في تمييز أنماط فن الخط العربي. ويعتبر الترميز التلقائي منهج تعلم عميق له القدرة علي اختزال حدود البيانات بالإضافة إلي استخلاص الملامح المميزة لفن الخط العربي. كما يمكن تصنيف الترميز الرقمي الي عدة طبقات، ويتشكل هذا النظام من ثلاث طبقات والتي تتكون من طبقتي تشفير لاستخلاص الملامح المميزة لفن الخط العربي وطبقة أخيرة لازمة لمرحلة التعرف. يمكن التعرف على الخط بشكل جماعي اعتمادًا علي الكلمات أو المقاطع الموجودة في الصور الخطية. تتكون مُدخلات النظام من كلمات مفردة أو صور مقسمة لمقاطع تشكل صورة الخط، وتمثل المُخرجات نوع الخط المتميز. تم تقييم هذا المنهج علي أساس مجموعة البيانات المحلية والعامة ودرجة نجاح في التعرف على الخط.