Intelligent Automation And Soft Computing, 2018
Copyright © 2018, TSI® Press
Vol.24,no.3,493–505

Highly Accurate Recognition of Handwritten Arabic Decimal Numbers Based on a Self-Organizing Maps Approach

[1,2]Amin Alqudah, [2]Hussein R. Al-Zoubi, [2,3]Mahmood A. Al-Khassaweneh, and [1]Mohammed Al-Qodah

[1]Department of Electrical Engineering, College of Engineering, Prince Sattam Bin Abdulaziz University, Al-Kharj, Saudi Arabia
[2]Computer Engineering Department, Hijjawi Faculty for Engineering Technology, Yarmouk University, Jordan.
[3]Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI, USA.

ABSTRACT
Handwritten numeral recognition is one of the most popular fields of research in automation because it is used in many applications. Indeed, automation has continually received substantial attention from researchers. Therefore, great efforts have been made to devise accurate recognition methods with high recognition ratios. In this paper, we propose a method for integrating the correlation coefficient with a Self-Organizing Maps (SOM)-based technique to recognize offline handwritten Arabic decimal digits. The simulation results show very high recognition rates compared with the rates achieved by other existing methods.

KEY WORDS: Correlation coefficient; reference image; recognition rate; SOM

1 INTRODUCTION

The electronics industry is currently transforming our traditional way of living. People now communicate using the Internet and mobile telephones and use digital devices in almost every aspect of their lives. Nevertheless, many daily activities and tasks still depend on classical methods of communication, such as using paper and a pencil or a pen. In many applications, automatic handwriting recognition can clearly save time and effort (Xiaofang, et. al., 2011; Khalil, et. al., 2017; Claudio, et. al., 2001; Gómez, et. al., 2001).

Automatic recognition of handwritten numbers is a field that has attracted many researchers to develop algorithms to increase recognition rates (Abirami and Murugappan, 2011; Ahmed and Moskowitz, 2004; Alhoniemi, 2002; Al-Omari and Al-Jarrah, 2004; Al-Zoubi, et. al., 2011; Alqudah, et. al., 2012). In this study, we employed the correlation coefficient (Ahmed and Moskowitz, 2004; Sun, et. al., 2003) and Self-Organizing Maps (SOMs) (Kaski, et. al., 1998; Kohonen, 1997; Kohonen, et. al., 1996; Simula and Kangas, 1995) for the offline recognition of handwritten decimal numbers and through an extended number of experiments demonstrated that very high recognition rates can be achieved. The measured features of Arabic numerals include angular span, distance span, horizontal span, and vertical span (Mahmoud, 2008).

In this section, we review the latest proposed methods in the field of numeral and character recognition with a focus on the recognition of handwritten numerals (Milson and Rao, 1976; Narasimhan, et. al., 1980). For example, the authors of (Rajput, et. al., 2010) proposed a method to recognize handwritten and printed Kannada numerals using crack codes and a Fourier descriptors template. The work of (Plamondon and Srihari, 2000) presented a survey of the most popular algorithms for preprocessing, character and word recognition, signature verification, writer authentication, and handwriting learning tools for online and offline recognition. In (Choudhary, et. al., 2011), the authors proposed a handwritten numeral recognition method using a modified Back Propagation Neural Network structure. In (Abirami and Murugappan, 2011), the identification of English numerals was conducted using rule-based classifiers and their sub-classifiers to implement a set of classification rules. This technique involves document vectorization, by which vectors are generated based on characters' shape, density, and

This paper was submitted July 20, 2017. Accepted: December 30, 2017

CONTACT Amin Alqudah a.alqudah@psau.edu.sa

transition features. The overall performance of this technique was observed to be 94%.

The authors of (Sas and Kaczmar, 2012) proposed a recognition method for writer-dependent handwriting. The proposed method continuously segments handwritten words using genetic algorithms. The goal is to improve the recognition accuracy by maximizing the similarity among images of the same character within subsets of images. The authors of (Zhao and Kit, 2011) used unsupervised learning to enhance the performance of supervised learning during Chinese-word segmentation. To integrate the strengths of the two methods, the authors applied various goodness-of-fit measures and were able to achieve an 8.96% reduction in error. In (Liu, et. al., 2011), a fast SOM clustering method was proposed to leverage the performance of text clustering systems while at the same time preserving comparable clustering quality. Unlike most term weighting approaches, which are based on document indexing, the authors of (Ren and Sohrab, 2013) proposed a class-indexing-based term weighting approach for automatic text classification (ATC). The experimental results indicated better classification than that achieved by existing classification methods.

Many methods that are used for the recognition of online and offline handwritten English and Arabic numerals are based on neural networks (Al-Omari and Al-Jarrah, 2004; Goltsev and Rachkovskij, 2005; Kherallah, et. al., 2008; Kussul and Baidyk, 2004; Lauer, et. al., 2007; Liu and Sako, 2006; Said, et. al., 1999) and support vector machines (Lauer, et. al., 2007; Soltanzadeh and Rahmati, 2004). Hidden Markov models have also been employed in the recognition of offline handwritten numerals (Mahmoud, 2008). Parkins and Nandi applied genetic programming to recognize handwritten digits (Parkins and Nandi, 2004). The authors of (Peng and Xu, 2012) proposed a classifier for pattern recognition called the Twin Mahalanobis distance-based Support Vector Machine (TMSVM), which represents a generalization of Twin Support Vector Machines (TSVMs) and other classification methods. TMSVM constructs two Mahalanobis distance-based kernels to optimize nonparallel hyperplanes using the covariance matrices of two classes of data. Lastly, the authors in (Al-Zoubi, et. al., 2011) proposed a global motion estimation method for the recognition of offline machine-printed Arabic digits.

Herein, we present a fast and accurate algorithm that exhibits very high recognition rates based on SOMs integrated with a correlation coefficient for the offline recognition of handwritten Arabic decimal numbers. For the numerals (1 – 9), we calculate the correlation coefficient between certain reference images generated by a SOM and the image under consideration. Recognition is performed by finding the reference image that has the highest correlation with the target image using the following equation:

$$Corr = \frac{\frac{1}{Ng}\sum_{i=1}^{Ng}(I_R(i) - mean(I_R))(I_T(i) - mean(I_T))}{\sigma_{I_R}.\sigma_{I_T}}, \quad (1)$$

where Ng is the number of pixels in the image, I_R and I_T are the reference and the target images, respectively, and σ_{I_R} and σ_{I_T} are the standard deviations of the reference and the target images, respectively. Moreover, we apply a special procedure for the recognition of the numeral 0, achieving recognition rates close to 100%, as discussed in Section 4.

This paper is organized as follows: a description of the preprocessing stage is presented in Section 2. Section 3 provides an introduction to SOM basics. In Section 4, we describe the proposed approach. SOM-classified images are shown in Section 5. The experimental results are presented in Section 6. Finally, Section 7 concludes the paper.

2 PREPROCESSING

IN offline recognition, we obtain a scanned image of the numeral to be recognized. The scanned image (original image) is converted to a gray-level image, which is then converted to a binary image. The binary image is cleaned using opening and closing transforms and through erosion and dilation to smooth the boundaries of the numeral to be recognized, maintain its size, and eliminate any noise. The original dimensions of the images used in this work are (32 × 32) pixels (including the numeral and the background). Because the numerals in images are different sizes, resizing is used to scale the numeral in the image to a standard size. All numerals (1 – 9) are resized to a common size by selecting the value of the nearest point that yields a piecewise-constant interpolant (http://www.mathworks.com) based on the nearest-neighbor interpolation algorithm. Figure 1 (a) shows the number “4” before preprocessing (image size of 32 × 32). Figure 1 (b) shows the number “4” with an image size of (20 × 15) pixels after the preprocessing step. The number has been scaled and shifted to the middle of the image.

The preprocessing step provides images with the same value a better chance to achieve a high correlation. For example, the images in Figures 1 and 2 represent the number “4”. The correlation between the images in Figure 1 (a) and Figure 2 (a) is 0.03, whereas the correlation between the numerals in the same images after shifting and scaling (shown in Figure 1 (b) and Figure 2 (b)) is 0.78. Deformation does not significantly affect the recognition rate. Rather, centering the scaled images provides a good chance for images that show the same numeral to have a high correlation value, which leads to higher recognition rates.

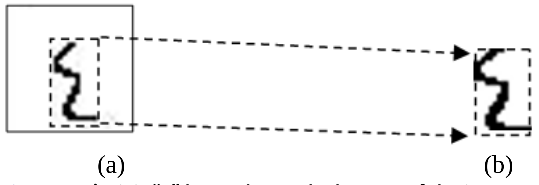


Figure 1. a) Digit "4" located near the bottom of the image (entire image size 32x32). b) Digit "4" centered in the middle of the image (image size 20x15).

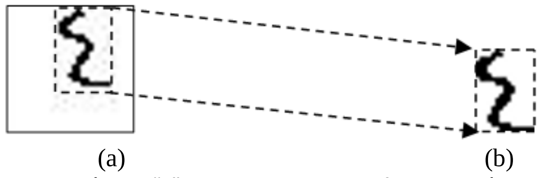


Figure 2. a) Digit "4" located near the top of the image (entire image size 32×32). b) Digit "4" centered in the middle of the image (image size 20× 15).

3 SELF-ORGANIZING MAPS (SOM'S)

CLASSIFYING handwritten numbers can help us extract useful information that can be used to discover new properties of handwriting and improve recognition rates. A self-organizing map is a type of neural network that is based on unsupervised learning. It performs a nonlinear mapping of data onto a two-dimensional map grid (Kohonen, 1997). SOMs have been used in different engineering applications (Kaski, et. al., 1998; Kohonen, 1997; Kohonen, et. al., 1996; Simula and Kangas, 1995; Alqudah and Al-Zoubi, 2015). A SOM consists of connected neurons, each of which is represented by a *d*-dimensional weight vector, where *d* is the dimension of the input vectors used for training. Figure 3 shows the SOM structure used in this study. Training the SOM requires updating the best matching unit (BMU) and its neighbors on the map, as shown in

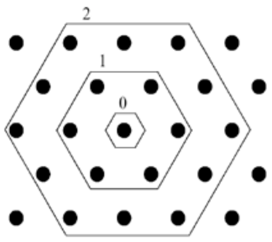


Figure 3. The hexagonal lattice structure of the SOM used in this study.

Figure 4, where the BMU and its neighbors are pulled toward the input sample, marked (×). The solid lines represent the connections before updating, whereas the dashed lines represent the connections after updating. At the end of training, the neurons will be organized into a structure that classifies the training samples into classes, where each neuron is the center of that class.

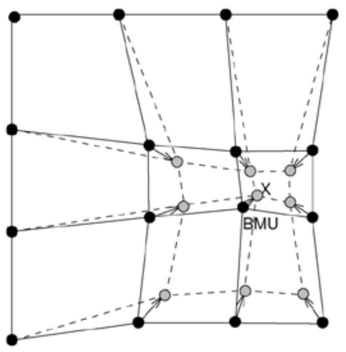


Figure 4. Updating the best matching unit (BMU).

Two training algorithms are generally used when training SOM networks: the sequential algorithm and the batch algorithm. Both algorithms provide the same mapping (Kohonen, 1993). In this work, we use the batch algorithm because of its performance in terms of speed. More information about SOM training can be found in many published studies (Alhoniemi, 2002; Kaski, et. al., 1998; Kohonen, 1997; Kohonen, et. al., 1996; Kohonen, 1993; Kohonen, 1991; Simula and Kangas, 1995; Vesanto, et. al., 2000).

4 THE PROPOSED APPROACH

THE proposed recognition approach consists of two stages: obtaining the reference images and recognition. The reference images are obtained by clustering a large number of handwritten numbers into clusters using SOMs and then drawing a reference image for every cluster. This image is created by taking the average of all of the images that belong to the same cluster. Averaging is performed using the following equation:

$$\mathrm{Im}_R(i,j) = \sum_{n=1}^{L} \mathrm{Im}_R^{(n)}(i,j)/L, \qquad (2)$$

where Im_R is the created reference image, (*i*,*j*) is the location of the pixel in the image, $\mathrm{Im}_R^{(n)}$ is the nth image in class *R*, and *L* is the number of images in class *R*.

We cluster/classify the images of each Arabic decimal number (1 – 9) into N clusters/classes, yielding a total of (9 $\times N$) clusters or reference images. Recognition is performed using the correlation coefficient between the image under consideration and all of the reference images.

The image is compared to all of the reference images and placed in the class with the highest correlation. Figure 5 illustrates the first stage of the proposed approach. It is worth mentioning that clustering the images of each decimal number (1 – 9) into N classes emphasizes the features that characterize how people write decimal numbers.

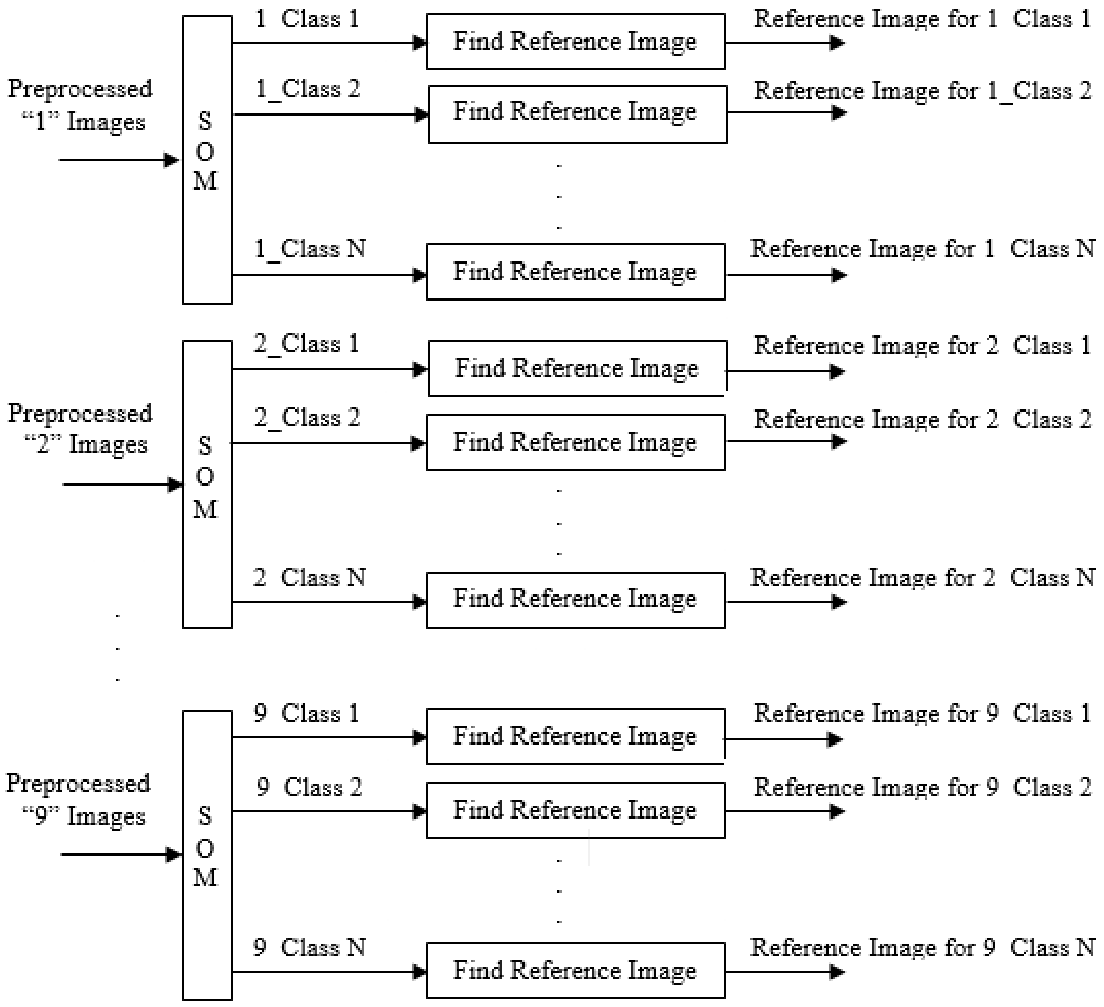

Preprocessed "1" Images
SOM
1_Class 1
1_Class 2
1_Class N
Find Reference Image
Reference Image for 1_Class 1
Reference Image for 1_Class 2
Reference Image for 1_Class N
Preprocessed "2" Images
2_Class 1
2_Class 2
2_Class N
Reference Image for 2_Class 1
Reference Image for 2_Class 2
Reference Image for 2_Class N
Preprocessed "9" Images
9_Class 1
9_Class 2
9_Class N
Reference Image for 9_Class 1
Reference Image for 9_Class 2
Reference Image for 9_Class N

Figure 5. **The steps followed to draw the reference images for numerals (1 – 9).**

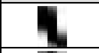
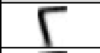
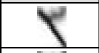
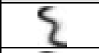
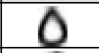
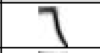
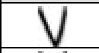
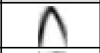
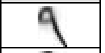
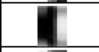
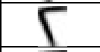
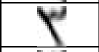
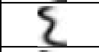
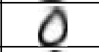
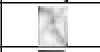
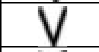
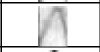
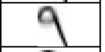
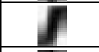
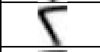
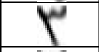
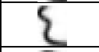
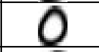
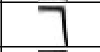
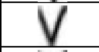
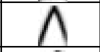
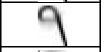
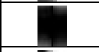
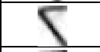
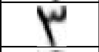
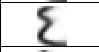
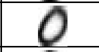
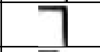
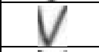
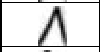
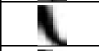
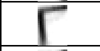
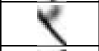
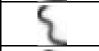
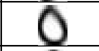
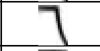
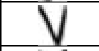
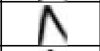
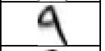
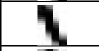
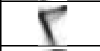
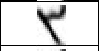
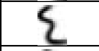
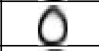
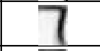
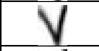
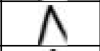
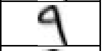
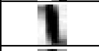
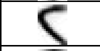
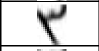
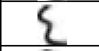
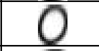
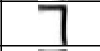
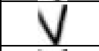
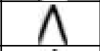
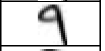
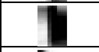
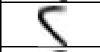
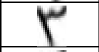
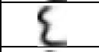
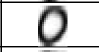
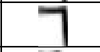
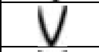
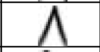
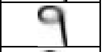
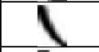
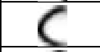
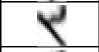
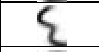
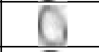
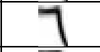
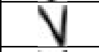
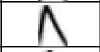
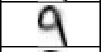
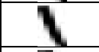
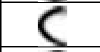
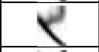
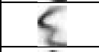
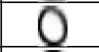
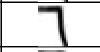
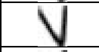
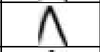
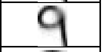
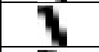
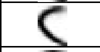
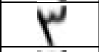
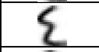
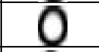
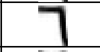
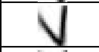
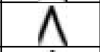
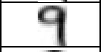
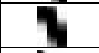
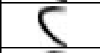
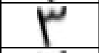
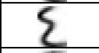
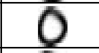
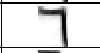
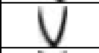
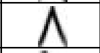
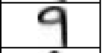
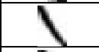
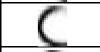
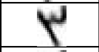
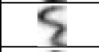
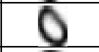
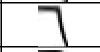
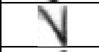
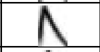
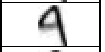
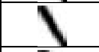
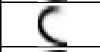
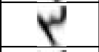
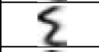
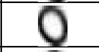
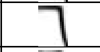
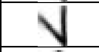
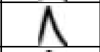
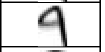
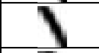
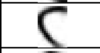
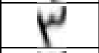
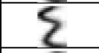
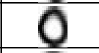
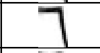
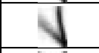
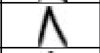
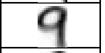
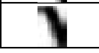
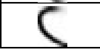
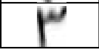
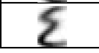
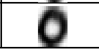
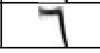
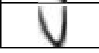
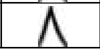
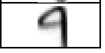
Furthermore, having one class for each decimal would hide many features of written decimals, especially uncommon ones. Therefore, having many classes for each number divides the numbers into groups, and uncommon features have a better chance of being shown.

To recognize a certain numeral, the number in the image is first centered and scaled to a (20 × 15) pixel image, as shown in the previous step. The correlations between this new image and the (9 × N) reference images are calculated, yielding (9 × N) correlation values, one value for each reference image. The number is classified as having the value of the image with the maximum correlation. For the numeral "0", a special procedure is applied in which the ratio between the height and the width of the tested numeral is calculated. If the ratio is in the range (0.5-2.0) and the ratio of white pixels to black pixels is very small (less than 0.5, ideally zero), then the numeral under test is classified as "0". Otherwise, the tested numeral undergoes the previously described process. With this method, we are able to achieve very high recognition rates. Preprocessing is applied after the "0" special case test to avoid scaling the numeral (if it is actually "0").

5 CLASSES

AS mentioned in Section IV, N reference images are constructed for each of the 9 numerals (1 – 9) corresponding to N classes. Each reference image is obtained by calculating the average of all of the images that belong to the same class. In this study, we test different values of N (1, 4, 9, 16, 25, and 49). Table 1 shows 144 reference images at N = 16. This table reveals the most common styles that people follow in handwriting Arabic numbers. It is worth mentioning that the images shown in Table 1 are created by classifying and averaging the classes of the

Table 1. Reference images for digits (1 – 9), N = 16.

Number/ Class	1	2	3	4	5	6	7	8	9
1									
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									

shifted and resized images of each number, which is why fuzzy images are observed, as in the case of decimal 6 class 2. In this study, a large number of people have participated by providing us with real handwritten documents containing words and numbers. The numbers are extracted with an image size of (32 × 32 pixels). Table 2 shows the 10 Arabic numerals and their equivalents in English numerals. We collect approximately 4,860 samples of each decimal, yielding 48,600 images for all decimal digits (0 – 9). Out of the 48,600 images, 30,000 images are used to train the SOM classifier (3000 images/numeral), and the rest are used for testing.

To obtain a better understanding of the data we are using, the frequency of each of the handwriting styles is calculated. Table 3 shows the frequency of the images in each class for digits (1 – 9) when the number of classes N = 16. From Table 3, it can be observed that the digit "6" has the largest deviation, 1.93, between different handwriting styles, whereas the digit "7" has the smallest deviation, 0.74. Table 4, which is based on information derived from Table 3, shows the images of the most and the least frequent classes. Moreover, the table shows interesting and useful statistics about the most and least frequent classes (the frequencies of these classes are shown in bold font in Table 3).

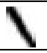
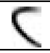
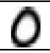
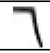
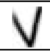
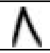
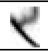
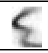
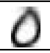
Table 2. Arabic numerals and their corresponding English numerals.

Printed Arabic Number Format	Corresponding Printed English Number Format
٠	0
١	1
٢	2
٣	3
٤	4
٥	5
٦	6
٧	7
٨	8
٩	9

Table 3. Frequency (%) of images in each class for digits (1 – 9), N = 16.

Number/ Class	1	2	3	4	5	6	7	8	9
1	7.73	5.90	6.16	7.19	4.93	7.56	6.96	5.13	6.36
2	5.03	7.60	5.90	6.93	**4.39**	**2.50**	6.26	**3.63**	8.33
3	4.63	6.26	8.00	6.56	**8.79**	6.90	6.03	7.33	7.80
4	8.16	5.36	7.10	5.33	5.03	5.06	6.10	5.26	**2.90**
5	4.83	4.86	5.50	7.63	7.10	8.33	5.86	5.70	5.70
6	7.43	**4.46**	5.70	**9.03**	8.03	2.53	**7.06**	5.06	7.70
7	**3.90**	5.80	7.10	7.86	7.70	6.90	6.06	6.96	8.36
8	6.33	6.90	7.93	6.13	5.76	4.33	6.96	5.83	6.16
9	6.80	4.90	5.09	7.13	4.53	6.60	6.46	7.06	5.50
10	7.43	5.76	**4.00**	**3.63**	7.33	7.56	5.93	6.90	**8.60**
11	6.36	7.30	**8.96**	6.66	6.10	5.66	7.03	6.20	4.93
12	6.00	**8.43**	6.93	6.73	7.23	5.46	6.73	6.66	6.23
13	5.83	5.13	5.56	4.93	4.43	**9.40**	5.53	6.73	5.73
14	**8.63**	7.96	5.30	5.73	6.56	7.43	6.90	6.96	6.40
15	6.36	6.20	6.03	4.30	7.00	7.76	**4.20**	**8.30**	4.10
16	4.50	7.13	4.70	4.16	5.03	5.96	5.86	6.23	5.16
Total	100	100	100	100	100	100	100	100	100
Average	6.25	6.25	6.25	6.25	6.25	6.25	6.25	6.25	6.25
Median	6.34	6.05	5.96	6.61	6.33	6.75	6.18	6.44	6.19
STDEV	1.40	1.19	1.32	1.47	1.41	**1.93**	**0.74**	1.12	1.60

Table 4. Most and least frequent classes for digits (1 – 9), N = 16.

Number	1	2	3	4	5	6	7	8	9
Most frequent handwriting class after preprocessing	14th	12th	11th	6th	3rd	13th	6th	15th	10th
Frequency	8.63	8.43	8.96	9.03	8.79	9.40	7.06	8.30	8.60
Average image									
Least frequent handwriting class	7th	6th	10th	10th	2nd	2nd	15th	2nd	4th
Frequency	3.90	4.46	4.00	3.63	4.39	2.50	4.20	3.63	2.90
Average image									

6 RESULTS AND DISCUSSION

THE approach described in the previous section is first tested using the 30,000 training images; as expected, the recognition rate is 100%, which certifies the sufficiency of the features considered. The same approach is also tested using 1,860 test images for each Arabic decimal number, yielding 18,600 test images. If the numeral in the image is not determined to be "0", it is centered and scaled to a (20 × 15) image as a preprocessing step to improve recognition. Then, the correlation between the test image (the centered and scaled numeral) and the reference images is calculated. The number value corresponding to the reference image with the highest correlation with the test image is assumed to be correct. As the most important step, all preprocessed training images are clustered using the SOM technique. The images are clustered into 1, 4, 16, 25, 36, and 49 classes. Each time, the recognition rate is calculated. We observe that the recognition rate increases with the number of classes. This relationship holds true until the number of classes reaches 16, when a recognition rate of 100% is achieved. This recognition rate remains constant until the number of classes reaches 25; after that point, the recognition rate starts to decrease, as shown in Figure 6 and Table 5.

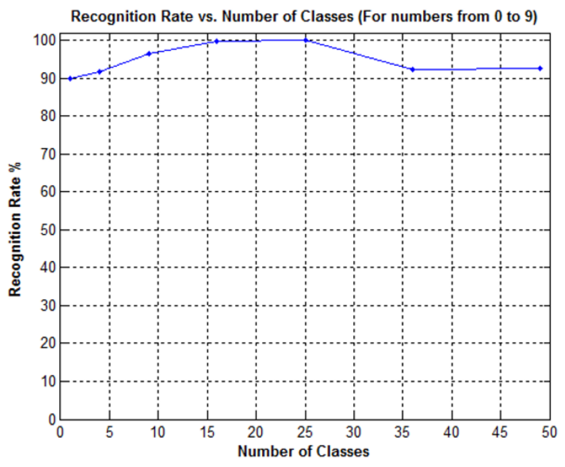


Figure 6. **Average recognition rate vs. number of classes. (The number "0" is recognized based on a special case test. The numbers (1 to 9) are recognized based on SOM).**

Table 5. **Effect of the number of classes on recognition rate and testing time, including the special case test for "0".**

Number of Classes	Recognition Rate (%)	Time to Test (seconds)
1	90.08	228
4	91.79	666
9	96.53	1602
16	99.99	2322
25	99.97	3587
36	92.43	5196
49	92.59	7009

The recognition rate is at least 90% for any number of classes. This behavior occurs because by increasing the number of classes, we increase the number of hidden features that are observed, which helps to increase the recognition rate. This trend holds true until the number of classes reaches 25, after which the recognition rate begins to decrease. This decrease occurs because having too many classes makes the recognition mechanism very sensitive to minor features, which leads to false recognitions. Furthermore, this degradation in performance could be related to the ratio of the reference image size to the number of images in any class. For example, when the number of classes is 49, the average number of images in each class is approximately 61 images/class (3000/49); the size of the reference image created from these images in this class, after averaging, is (20 × 15=300) pixels. This large image size (300 pixels), when compared to the size of the 61 images/classes, suggests that each pixel has only a small chance to be represented. Each pixel has a recognition probability of 61/300, which is minor compared to that observed when the number of classes is 16. In that case, the ratio is approximately 187.5/300. The latter case results in a representation that is approximately 3 times better; therefore, we focus on the case of N = 16 classes.

6.1 *Effect of reducing the number of subclasses used for recognition*

We select the case involving 16 classes for further study. We obtain the frequency of selection for each class. The results are shown in Table 6. As shown in the table, it is clear that there are dominant classes for each number. For example, for the decimal number "1", class 11 is selected more than 99% of the time, whereas for the decimal number "6", class 3 and class 4 are selected at a similar rate. Therefore, the most and the second most selected class for each number are studied separately. (These classes are indicated in bold in Table 6).

Table 7 shows the most and the second most selected for digits (1 – 9) for the case of N = 16 classes. The table shows the representative image for each class as well as the selection percentage. If we focus on the most selected class for each number, it is observed that 67.27% of the correct recognitions are derived from this particular class. Furthermore, if we focus on the second most selected class, it is observed that 26.19% of the correct recognitions are derived from this class. Combining both classes and ignoring the other classes would give us a recognition rate of approximately 93.46%. This high rate implies that this system could be made even simpler if we consider only these two classes, in which case we would still have a considerable recognition rate.

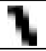
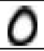
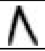
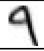
6.2 *Time requirements*

In addition to the recognition accuracy, a system with a high recognition rate should also take into consideration the time requirements for testing. Therefore, we measure the time required to test 1,860 images for each digit. This work is carried out using MATLAB version 7.8 (R2009a) on a computer running Windows Vista (64-bit) with 4 GB of RAM and a processor with a 2-GHz clock rate. Table 5 lists the recorded average recognition rate versus the test time when 1, 4, 9, 16, 25, 36, and 49 classes are used (includes the special test for "0"). As expected, longer test times are needed when more classes are used. Consequently, the tradeoff between accuracy and performance should be determined. In the rest of this

Table 6. Correct recognition percentage for each class for digits (1 – 9), N = 16.

Number/ Class	1	2	3	4	5	6	7	8	9
1	0	**62.58**	0	0.05	0.05	0	6.13	0	0
2	0.05	**20.81**	0	0	0	0	**11.02**	0	2.74
3	0	0	1.94	0	**79.68**	**50.81**	0	0	0
4	0	0	0.11	0	0.05	**49.09**	0	0	0
5	0	0	0	**33.98**	0	0	0	0	0.27
6	**0.43**	0	1.4	0	0.91	0	0.05	0	**52.69**
7	0	0	**67.37**	**48.87**	0	0	0	3.66	0.05
8	0	0	0.22	0	0	0.05	0	**32.9**	0
9	0	0	0	13.06	0	0	0	0	0.38
10	0.05	0	0.32	0.05	0	0	0.65	0	0
11	**99.03**	0	0	0	1.72	0	0	0	0
12	0.16	0.22	0	3.98	**17.53**	0	0	0	**41.34**
13	0.22	16.29	**28.66**	0	0	0	0	0	0
14	0	0.11	0	0	0	0	**81.29**	0.32	2.53
15	0	0	0	0	0	0.05	0	**63.12**	0
16	0	0	0	0	0	0	0.86	0	0
Total	99.94	100.0	100.0	100.0	99.94	100	100	100	100

Table 7. Highest and second highest correct recognition percentage for digits (1 – 9), N = 16.

Number	1	2	3	4	5	6	7	8	9
Class number of highest correct recognition	11th	1st	7th	7th	3rd	3rd	14th	15th	6th
Recognition rate (RC1 %)	99.03	62.58	67.37	48.87	79.68	50.81	81.29	63.12	52.69
Average image									
Class number of second highest correct recognition (RC2 %)	6th	2nd	13th	5th	12th	4th	2nd	8th	12th
Recognition rate %	0.43	20.81	28.66	33.98	17.53	49.09	11.02	32.9	41.34
Average image									
RC1+RC2 (%)	99.46	83.39	96.03	82.85	97.21	99.9	92.31	96.02	94.03

section, the case of N = 16 classes will be considered. We will discuss methods for further improving the performance of the proposed recognition system.

6.3 Case study: Number of classes N=16

To obtain a profound understanding of the recognition behavior of our proposed method, we will closely examine the results that we have obtained. Table 8 shows the recognition rate achieved for each digit with 16 classes. At a glance, we can notice the high accuracy of the proposed recognition approach, for which an average recognition ratio of 99.99% is achieved, with a 100% recognition ratio for the digits 0, 2, 3, 4, 6, 7, 8, and 9.

Table 6 presents the correct recognition percentages achieved for each class for digits (1 – 9) with 16 classes. Table 7 highlights the highest and the second highest recognition rates (i.e., successful classes) for the digits (1 – 9). The average reference image is also shown in the table. Samples of some of the successfully recognized numbers can be found in Table 9. The images shown in the table are the original scans obtained before preprocessing.

Table 8. Recognition rates for each decimal digit, N = 16. (Number "0" was recognized based on a special case test).

Decimal digit	Correct recognition rate (%)
Digit 0	100
Digit 1	99.94
Digit 2	100
Digit 3	100
Digit 4	100
Digit 5	99.94
Digit 6	100
Digit 7	100
Digit 8	100
Digit 9	100
Average	99.99

On the other hand, Table 10 shows all of the misrecognized numbers. The table shows the image index, the original image, the actual digital value, the misrecognized value, the misclassified class, and the reference image for that class in each case. The table shows that the digits "1" and "5" have been misrecognized only once each as "0" and "2", respectively.

6.4 System performance when using 16 classes out of 16 vs. when using 2 classes out of 16

A closer look at Table 6 reveals that the majority of the classes achieve a 0% or very poor recognition rate (in other words, Table 6 is sparse). Therefore, in a trial to obtain better performance using our proposed system, we investigate a system using only the most and second most successful classes for each of the decimal digits. That is, rather than using all of the 16 classes, why not use only 2? Table 11 shows the recognition rates and the test times obtained when all of the 16 classes are used and when only the two most

Table 9. Examples of images of correctly recognized numbers, N = 16. (The number "0" was recognized based on a special case test).

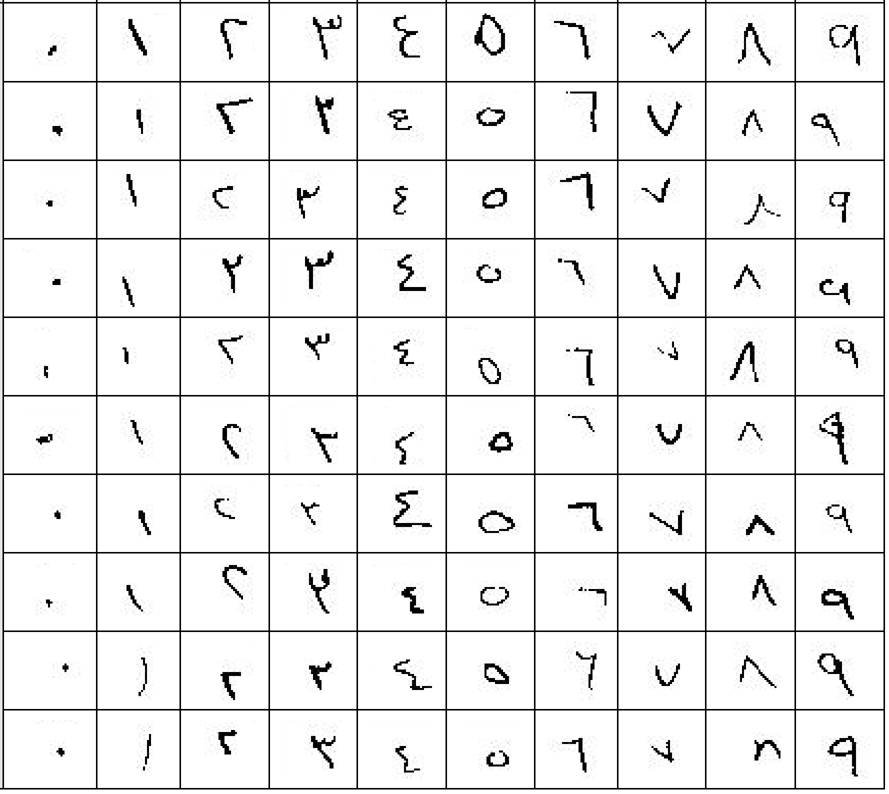
Arabic Number format	٠	١	٢	٣	٤	٥	٦	٧	٨	٩
Corresponding English Number format	0	1	2	3	4	5	6	7	8	9
Examples of successfully recognized Arabic numbers										

Table 10. The incorrectly recognized numbers.

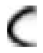
Image Index	Original Image	Preprocessed Image	Actual Digit Value	Classification Based on 16 classes		
				Misrecognized As	Class	Class Reference Image
1			1	0	1	
2			5	2	9	

Table 11. Comparison of recognition rates and test times when all of the 16 classes are used and when only the two most successful of the 16 classes are used.

Decimal digit	2 classes: Recognition Ratio (%)	16 classes: Recognition Ratio (%)	2 classes: Time to Test 1860 images (seconds) T1	16 classes: Time to Test 1860 images (seconds) T2	Speedup (T2/T1)
Digit 1	100	99.94	47	398	8.46
Digit 2	100	100	47	400	8.51
Digit 3	100	100	50	399	7.98
Digit 4	99.90	100	48	399	8.31
Digit 5	100	99.94	48	395	8.22
Digit 6	100	100	48	396	8.25
Digit 7	100	100	48	397	8.27
Digit 8	100	100	47	405	8.61
Digit 9	100	100	48	403	8.39
Average	99.99	99.99	47.9	399.1	8.33

successful of the 16 classes are used. Surprisingly, it can be observed that the average recognition rates are the same in both cases (99.99%). Looking at the recognition rate for each digit individually, other than that for the digit “4”, the recognition rates are either enhanced or remain the same.

Table 12 shows all of the misrecognized numbers based on 2 classes among the 16 classes. The table shows that for the digit “4”, two misrecognized pictures appear; the recognition rate decreases by 0.10% (from 100% to 99.90%).

To explain the changes in the achieved recognition rate when the number of classes is reduced to 2 from 16, let us look at Table 13. The table lists all of the misrecognized numbers when using 2 classes out of 16 and when using all 16 classes. Table 13 shows that the incorrectly assigned classes are removed from the set for the case in which only 2 classes are used.

These results demonstrate why the recognition rates for the digits “1” and “5” increased while the recognition rate for the digit “4” decreased: in the 2-class case, the 11th and 6th classes were selected for the digit “1”, the 7th and 5th classes were selected for the digit “4”, and the 3rd and 12th classes were selected for the digit “5”. To better illustrate this point, let us take the 2nd row of Table 13 as an example. This row clarifies what has happened for the image with Index 1. This image, with an actual digital value 4, was correctly recognized as “4” when the number of classes was 16; Class number 1 of the digit “4” showed the highest correlation, 0.5232. In the 2-class case, Class number 1 of the digit “4” was removed because it was not one of the most successful classes. Therefore, Class number 2 of the digit “7”, which remained in the set, exhibited the next highest correlation, 0.3516, and was selected; the number “4” was misrecognized as the number “7”.

In terms of performance, an average increase in speed of 8.33 is achieved when 2 classes are used rather than 16, as indicated in Table 11, with the same recognition accuracy. The same table also shows that the average test time needed to recognize 1,860 images is 47.9 seconds, which means that an average of 26 milliseconds is needed to recognize a number. Thus, in addition to exhibiting high accuracy, the proposed system is also fast and can be implemented in online applications.

7 CONCLUSIONS AND FUTURE WORK

IN this study, we propose a method that integrates the use of correlation coefficients with that of SOMs for the recognition of handwritten Arabic decimal numbers. The proposed recognition approach has three main advantages. First, the method is simple: the proposed approach uses 2 basic methods, the correlation coefficient method and the SOM method,

Table 12. Incorrectly recognized numbers based on 2 out of the 16 classes.

Image Index	Original Image	Preprocessed Image	Actual Digit Value	Classification Based on 2 classes		
				Misrecognized As	Class	Class Reference Image
1			4	1	6	
2			4	1	6	

Table 13. All of the misrecognized images, N = 16. Case 1: 16 out of the 16 classes are used. Case 2: 2 out of the 16 classes are used.

Image Index	Origina l Image	Preprocessed Image	Actual Digit Value	Classification based on 16 Classes				Classification based on 2 classes			
				Valu e	Class	Image	Corr.	Value	Class	Image	Corr.
1			1	0	1		0.7220	1	6		0.6378
1			4	4	1		0.5232	7	2		0.3516
2			4	4	10		0.6910	1	6		0.4242
2			5	2	9		0.4552	5	3		0.4306

both of which are known to be simple and fast. Second, the method is highly accurate: with the proposed method, the results are nearly 100% accurate. Third, the method exhibits high performance: the proposed approach performs recognition in only a few milliseconds. In future work, we plan to apply this approach for the recognition of handwritten Arabic and English characters.

ACKNOWLEDGMENT

THIS project was supported by the Deanship of Scientific Research at Prince Sattam Bin Abdulaziz University under the research project # 2017/01/7551.

DISCLOSURE STATEMENT

NO potential conflict of interest was reported by the authors.

REFERENCES

S. Abirami and S. Murugappan, (2011). Scripts and Numerals Identification From Printed Multilingual Document Images, *Computer Science and Information Technology*. 1, 129-146.

F. Ahmed and S. Moskowitz, (2004). Correlation-based watermarking method for image authentication applications, *Optical Engineering*. 43(8). 1833-1838.

E. Alhoniemi. (2002). Unsupervised pattern recognition methods for exploratory analysis of industrial process data, doctoral thesis, *Helsinki University of Technology*.

F. Al-Omari and O. Al-Jarrah, (2004). Handwritten Indian Numerals Recognition System Using Probabilistic Neural Networks, *Advanced Engineering Informatics*. 18(1), 9-16.

A. Alqudah and H. Al-Zoubi, (2015). Efficient k-Class Approach for Face Recognition, *Computers and Electrical Engineering*. 45, 260-273.

A. Alqudah, H. Al-Zoubi, and M. Al-Khassaweneh, (2012). Shift and Scale Invariant Recognition of Printed Numerals, *Journal of Abhath Al-Yarmouk for Basic Science and Engineering*. 21(1), 41-49.

H. Al-Zoubi, M. Al-Khassaweneh, and A. Alqudah, (2011). Precise and Accurate Decimal Number Recognition Using Global Motion Estimation, *International Journal of Artificial Intelligence and Soft Computing (IJAISC)*. 2(4), 287–301.

A. Choudhary, R. Rishi, and S. Ahlawat, (2011). Handwritten Numeral Recognition Using Modified BP ANN Structure, *Advanced Computing*. 133, 56-65.

C. De Stefano, A. Iuliano, and A. Marcelli, (2001). A Shape-Based Algorithm for Detecting Ligatures in On-Line Handwriting, *Intelligent Automation & Soft Computing*. 7(3), 187-194.

A. Goltsev and D. Rachkovskij, (2005). Combination of the Assembly Neural Network with a Perceptron for Recognition of Handwritten Digits Arranged in Numeral Strings, *The Journal of the Pattern Recognition Society*. 38(3), 315-322.

E. Gómez, Y.A. Dimitriadis, M. Sánchez-Reyes Más, P. Sánchez Gracía, J.M. Cano Izquierdo and J. López Coronado, (2001). On-Line Character Analysis and Recognition With Fuzzy Neural Networks, *Intelligent Automation & Soft Computing*. 7(3), 163-175.

http://www.mathworks.com/help/images/ref/imresize.html

S. Kaski, S., J. Kangas, and T. Kohonen, (1998). Bibliography of self-organizing map (SOM) papers: 1981–1997, *Neural Computing Surveys.* 1, 102–350.

K. El Hindi, M. Khayyat and A. Abu Kar, (2017). Comparing the Machine Ability to Recognize Hand-Written Hindu and Arabic Digits, *Intelligent Automation & Soft Computing*. 23(2), 295-301.

M. Kherallah, L. Haddad, A. Alimi, and A. Mitiche, (2008). On-line Handwritten Digit Recognition Based on Trajectory and Velocity Modeling, *Pattern Recognition Letters.* 29(5), 580-594.

T. Kohonen, (1991). Self-organizing maps: Optimization approaches, in Artificial Neural Networks, *The Netherlands: Elsevier.* 981–990.

T. Kohonen, (1993). Things you haven't heard about the self-organizing map, *Proc. ICNN* (1993). 1147-1156.

T. Kohonen, (1997). Self-Organizing Maps, *Springer*, 3rd edition.

T. Kohonen, E. Oja, O. Simula, A. Visa, and J. Kangas, (1996). Engineering applications of the self-organizing map, *Proc. of the IEEE* 84. 10, 1358–1384.

E. Kussul and T. Baidyk, (2004). Improved Method of Handwritten digit Recognition Tested on MNIST Database, *Image and Vision Computing.* 22(12), 971-981.

F. Lauer, C. Suen, and G. Bloch, (2007). A Trainable Feature Extractor for Handwritten Digit Recognition, *The Journal of the Pattern Recognition Society*. 40(6), 1816-1824.

X. Li, Y. Sun, M. Tang, X. Yan, and Y. Kang, (2011). A Neural Network-Based Intelligent Image Target Identification Method and Its Performance Analysis, *Intelligent Automation & Soft Computing*. 17(7), 885-896.

C.-L. Liu and H. Sako, (2006). Class-Specific Feature Polynomial Classifier for Pattern Classification and its Application to Handwritten Numeral Recognition, *The Journal of the Pattern Recognition Society*. 39(4), 669-681.

Y.-C. Liu, C. Wu, and M. Liu, (2011). Research of fast SOM clustering for text information, *Expert Systems with Applications*. 38(8), 9325–9333.

S. Mahmoud, (2008). Recognition of Writer-Independent Off-Line Handwritten Arabic (Indian) Numerals Using Hidden Markov Models, *Signal Processing.* 88(4), 844-857.

T. E. Milson and K. R. Rao, (1976). A Statistical Model for Machine Print Recognition, *IEEE Transactions on Systems, Man, and Cybernetics, SMC* 6. 10, 671-678.

M. A. Narasimhan, V. Devarajan, and K. R. Rao, (1980). Simulation of Alphanumeric Machine Print Recognition, *IEEE Transactions on Systems, Man, and Cybernetics, SMC* l0. 5, 270-275.

A. D. Parkins and A. K. Nandi, (2004). Genetic Programming Techniques for Hand Written Digit Recognition, *The Journal of Signal Processing* 84. 12, 2345-2365.

X. Peng and D. Xu, (2012). Twin Mahalanobis distance-based support vector machines for pattern recognition, *Information Sciences*. 200, 22–37.

R. Plamondon and S. N. Srihari, (2000). On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 22(1), 63-84.

G. R. Rajput, R. Horakeri, and S. Chandrakant, (2010). Printed and Handwritten Kannada Numeral Recognition Using Crack Codes and Fourier Descriptors Plate, *International Journal of Computer Applications.* 1(1), 53-58.

F. Ren, and M. Sohrab, (2013). Class-indexing-based term weighting for automatic text classification, *Information Sciences.* 236, 109–125.

F. N. Said, R. A. Yacoub, and C. Y. Suen, (1999). Recognition of English and Arabic Numerals Using a Dynamic Number of Hidden Neurons, *Proceedings of the Fifth International Conference on Document Analysis and Recognition*. 237-240.

J. Sas and U. Markowska-Kaczmar, (2012). Similarity-based training set acquisition for continuous handwriting recognition, *Information Sciences*. 191, 226–244.

O. Simula and J. Kangas, (1995). Process monitoring and visualization using self organizing maps, *In A. B. Bulsari, editor, Neural Networks for Chemical Engineers.* 371–384.

H. Soltanzadeh and M. Rahmati, (2004). Recognition of Persian handwritten digits using image profiles of multiple orientations, *Pattern Recognition Letters.* 25(14), 1569-1576.

S. Sun, H. Park, D. R. Haynor, and Y. Kim, (2003). Fast template matching using correlation-based adaptive predictive search, *International Journal of Imaging Systems and Technology.* 13(3), 169 – 178.

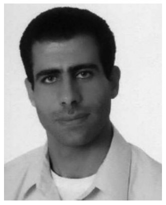
J. Vesanto, J. Himberg, E. Alhoniemi, and J. Parhankangas, (2000). SOM Toolbox for Matlab 5, Tech. Rep. A57, *Helsinki University of Technology*.

H. Zhao and C. Kit, (2011). Integrating unsupervised and supervised word segmentation: The role of goodness measures, *Information Sciences*. 181(1), 163–183.

NOTES ON CONTRIBUTORS

Amin Alqudah received his M.Sc. and Ph.D. in Electrical Engineering from the University of Colorado, USA in 2005, and from Colorado State University, USA in 2009, respectively. He received his bachelor degree in Communications Engineering from Yarmouk University, Jordan in 1999. Since 2009, he has been working with the Department of Computer Engineering, Hijjawi Faculty for Engineering Technology, Yarmouk University, Jordan. He has been working as an Associate Professor in the Electrical Engineering Department at Prince Sattam Bin Abdulaziz University, Al-Kharj, Saudi Arabia since Fall 2015. His research interests include Image Processing, Neural Networks, Machine Learning and adaptive signal processing.

Hussein R. Al-Zoubi received his MSE and Ph.D. in Computer Engineering from the University of Alabama in Huntsville, USA in 2004 and 2007, respectively. He received his bachelor degree in Electrical Engineering from Jordan University of Science and Technology. Since 2007, he has been working with the Department of Computer Engineering, Hijjawi Faculty for Engineering Technology, Yarmouk University, Jordan. He is currently a Professor. His research interests include machine vision, pattern recognition, image processing, computer networks and their applications: wireless and wired, security, multimedia, queuing analysis, and high-speed networks.



Mahmood A. Al-Khassaweneh received the B.S. degree in electrical engineering from Jordan University of Science and Technology, Jordan, in 2000, and the M.S. and the Ph.D. degrees in electrical engineering from Michigan State University, USA, in 2003 and 2007, respectively. Since Fall 2007, he has been working with the Department of Computer Engineering, Hijjawi Faculty for Engineering Technology, Yarmouk University, Jordan. He has been working as a Visiting Associate Professor in the Electrical and Computer Engineering Department at Michigan State University, East Lansing, MI, USA, since Fall 2014. His research interests include Image Processing, Computer Vision, Medical Imaging, Data Security and applying signal processing methods to the brain signals.

Mohammed Al-Qodah received his Bachelor and M.Sc. in Electrical Engineering from the Jordan University of Science and Technology, Jordan in 2004 and 2008, respectively. Since 2009, he has been a Lecturer with the Prince Sattam Bin Abdulaziz University, Al-Kharj, Saudi Arabia. His research interests include information theory, nonlinear and chaos theory and image processing.