Signal Processing 140 (2017) 87–96

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Image encryption method based on chaotic fuzzy cellular neural networks

K. Ratnavelu[a], M. Kalpana[a], P. Balasubramaniam[b], K. Wong[c,*], P. Raveendran[d]

[a] Institute of Mathematical Sciences, Faculty of Science, University of Malaya, Malaysia
[b] Department of Mathematics, Gandhigram Rural Institute - Deemed University, India
[c] School of Information Technology, Monash University Malaysia, Malaysia
[d] Faculty of Engineering, University of Malaya, Malaysia

ARTICLE INFO

Article history:
Received 28 December 2016
Revised 29 April 2017
Accepted 2 May 2017
Available online 11 May 2017

Keywords:
Chaos
Encryption
Leakage delay
Fuzzy cellular neural network

ABSTRACT

In this work, an image encryption method is proposed based on fuzzy cellular neural network (FCNN). First, the shortcomings of FCNN in encrypting image are identified, and the FCNN model is then modified to address these shortcomings. Specifically, a theoretical framework is developed to identify the values of the parameters of FCNN to generate chaotic signals, which are in turn utilized to encrypt the image. The encryption method is designed where an encrypted pixel is generated based on the corresponding plaintext pixel together with the neighbouring encrypted pixels. The proposed method has a key sensitivity in the order of 10^{-10} to achieve adequate security robustness. Further evaluations on standard test images verified and confirmed that the proposed encryption method is robust against plaintext-only (i.e., brutal force) and chosen-plaintext attacks.

1. Introduction

Image encryption is a process that transforms an image into an unintelligible form to mask its perceptual semantic [1,2]. With the advent of affordable capturing devices, ubiquitous network connection and free social networking service, images are generated and shared online at a staggering rate of 300 million pictures per day on Facebook itself [3]. Free cloud storage services further encourage users to store personal images online for easy of access. However, even ordinary users start to concern about their privacies as well as safeties nowadays. Therefore, encryption emerges as an important layer of protection for both privacy and safety in the age of cloud computing.

Although the conventional encryption methods such as IDEA, AES, RSA and DES can be deployed directly to encrypt image (e.g., treating image as a sequence of values), these methods are insufficient for encryption as detailed in Ref. [4].

Chaos appears as an attractive alternative for encryption because of its complex dynamic behaviors, which shows apparently random occurrences within a determined nonlinear system or process. Also, chaos has the characteristics of categories of noise-sensitive initial long-term unpredictability, ergodicity, and the divergence index, which are the desired properties in the application of encryption. Therefore, many digital image encryption methods have been proposed based on chaotic systems [5–10]. For example, Zhou et al., have invented a novel image encryption algorithm based on chaos and line map [5], while Assad and Farajallah proposed a new chaos-based image encryption system [8]. The problem of cracking a hierarchical chaotic image encryption algorithm based on permutation has been derived by Li, [10].

In addition, neural networks possess attractive properties such as high nonlinearity, parameter sensitivity and learning ability, hence they are widely utilized as an alternative choice for information protection, such as data encryption, data authentication and intrusion detection [11]. As a results, various chaotic neural based encryption algorithms are proposed [12–15]. For example, Wen et al. investigated into the problem of global exponential lag synchronization of a class of switched neural networks with time-varying delays via the neural activation controller and its applications in image encryption [12]. Chua and Yang [16,17] then proposed a new model-traditional cellular neural networks (CNNs). Subsequently, CNNs have attracted much attention because of their established background in theory and practical applications, including image processing.

Based on traditional CNN, Yang et al. proposed the fuzzy cellular neural networks (FCNNs) [18,19], which integrates fuzzy logic

* Corresponding author.
E-mail addresses: kururatna2012@gmail.com (K. Ratnavelu), kalpana.nitt@gmail.com (M. Kalpana), balugru@gmail.com (P. Balasubramaniam), wong.koksheik@monash.edu, koksheik.um@gmail.com (K. Wong), ravee@um.edu.my (P. Raveendran).

http://dx.doi.org/10.1016/j.sigpro.2017.05.002
0165-1684/

into the structure of traditional CNN and maintains local connectedness among cells. FCNNs is a generalization of CNNs with high level information processing capability, such as image understanding of fuzzy systems. Therefore, it is of great importance to analyze the dynamical behaviors of FCNNs both in theory and applications. Recently, Abdurahman et al. [20] investigated into the theoretical results of finite-time synchronization for FCNNs with time-varying delays. However, to the best of our knowledge, there is no results reported for FCNNs in the field of encryption.

To fill the research gap, in this paper we reformulate the FCNN system for encryption purpose by making it more secure against chosen-plaintext attack. Specifically, to generate an encrypted pixel, both the pixel being processed and those that are already encrypted are considered. This paper makes the following contributions: (a) The key space is large enough to make brute-force attacks infeasible; (b) a quantitative analysis with comparative result is provided, and (c) Experimental results are shown to prove the effectiveness of our method.

The rest of this paper is organized as follows. In Section 2 presents the background study of chaotic FCNN. Section 3 reformulates the chaotic FCNN for image encryption purpose. Experiments results are presented in Section 4. Analysis including key sensitivity and statistics are discussed in Section 5. Finally, Section 6 concludes this paper.

2. Background study

Similar to neural networks, CNNs are a parallel computing paradigm with the difference that communication is allowed only between neighboring units [16,17]. Its application include image processing, associative memories, classification of patterns, analyzing 3D surfaces, quadratic optimization, solving partial differential equations, reducing non-visual problems to geometric maps, modeling biological vision and other sensory-motor organs, and so forth.

There are various uncertainties in every stage of image processing, for example, loss of information when 3D objects are projected into 2D plane, additive and non-additive noise in the transmission processes, lack of quantitative measurement of image quality, imprecision in computations, etc. Fuzzy set theory provides the mathematical tools to capture these uncertainties. Therefore, it is reasonable to integrate fuzzy set theory into the CNN paradigm to give birth to a new image-processing paradigm termed FCNN, which takes uncertainties into consideration. Each cell in an FCNN contains fuzzy operating abilities, yet the entire network is governed by the cellular computing laws as proposed by Yang et al. [18,19]. FCNN has been proven to be a very useful paradigm for image processing and pattern recognition.

Consider the following chaotic FCNN:

$$\begin{cases} \dot{x}_i(t) = -d_i x_i(t-\sigma) + \sum_{j=1}^{n} a_{ij} f_j(x_j(t)) + \sum_{j=1}^{n} b_{ij} f_j(x_j(t-\tau(t))) \\ \qquad + B_i + \bigwedge_{j=1}^{n} \alpha_{ij} \int_{-\infty}^{t} k_j(t-s) f_j(x_j(s)) ds \\ \qquad + \bigvee_{j=1}^{n} \beta_{ij} \int_{-\infty}^{t} k_j(t-s) f_j(x_j(s)) ds, \\ x_i(s) = \phi_i(s), \quad s \in (-\infty, 0], \quad i \in \{1, 2, \ldots, n\}, \end{cases} \tag{1}$$

where $\phi_i(\cdot) \in C((-\infty, 0], \mathbb{R})$ is the initial condition of FCNNs system (1); α_{ij} and β_{ij} are the elements of fuzzy feedback MIN and MAX templates, respectively; a_{ij} and b_{ij} are the elements of feedback template; $\bigwedge$ and $\bigvee$ denote the fuzzy AND and fuzzy OR operations, respectively; x_i denotes the state vector in FCNNs system (1) of the ith neuron; B_i denotes the external input of the ith neuron; $\sigma > 0$ is a constant which denotes the leakage delay; a_{ij} and b_{ij} denote the connection weights of the feedback template; d_i is a diagonal matrix, which represents the rate at which the ith neuron resets its potential to the resting state in isolation when disconnected from the networks and external inputs; f_j represents the neuron activation function; $k_j(s) \geq 0$ is the feedback kernel, which satisfies

$$\int_0^{\infty} k_j(s) ds = 1, \quad j \in \{1, 2, \ldots, n\}. \tag{2}$$

FCNN (1) generates chaotic signals, which can be applied to the pixels of an image to generate unintelligible encrypted image. However, in its unmodified form, FCNN falls short in two aspects: (i) key-sensitivity, where any slight change in the parameters (i.e., secret keys) of FCNN, there is no significant change in any of the RGB channels of the encrypted color image, and; (ii) plaintext sensitivity, where any slight change in color plain image will lead to slight (i.e., insignificant) change in the corresponding color pixel in the encrypted image. As a result, in its present form, FCNN based encryption is vulnerable to chosen-plaintext attack. In this work, we begin by showing how FCNN can be used in encrypting an image and further changes are needed to be made in the FCNN expressed in (1) to address the aforementioned issues, i.e., withstanding well-accepted cryptographical attacks.

3. System formulation

In this formulation, the variable t in the chaotic signal $\dot{x}_i(t)$ takes the values in the range of [1, MN], where $M \times N$ represents the dimension of the image. Further, to ease the presentation, the image $I(x, y)$ is linearized to a vector $IMG(K)$, where $K = 1, 2, 3, \cdots, M \times N$ for the image of dimension $M \times N$. To cater for color image, $IMG_i(K)$ refers to the ith color channel of the Kth pixel, where $i \in \{R, G, B\}$ in which R, G and B represent the red, green, and blue color channels, respectively. The goal is to use each generated FCNN chaotic signal expressed in (1) together with each pixel of the original image to generate the output encrypted pixel. Hence, the length of the variables t and K is the same. To add another layer of security, the chaotic signal $\dot{x}_i(t)$ can begin at any point in the originally generated chaotic signal, i.e., $t \in [t_0 + 1, t_0 + MN]$ for $t_0 \geq 0$. The new chaotic signal that can be used for image encryption, which is denoted by $y_i(K)$ and can be computed as

$$y_i(K) = \mathrm{mod}\Big(IMG_i(K) + c\dot{x}_i(K), 256\Big). \tag{3}$$

where $c = 10^{11}$, and $\mathrm{mod}\ (\cdot, 256) \in [0, 255]$ refers to the remainder of the division by 255. Now to generate the encrypted image, each new chaotic signal $\dot{y}_i(K)$ is multiplied with each original pixel $IMG(K)$. This can be achieved by

$$E(IMG(K)) = IMG(K) \times y_i(K), \tag{4}$$

where $E(IMG(K))$ is the encrypted pixel of $IMG(K)$.

However, when any parameter in (1), i.e., a_{ij}, b_{ij}, d_{ij}, α_{ij}, and β_{ij}, is slightly changed, then it only affects the channel where its parameter was changed. Hence, the encryption method has low key-sensitivity. To overcome the abovementioned drawback, we introduce further substantial changes in (4) in a way that withstands the well-accepted cryptographical attacks.

First, we introduce the key E_{key} that depends on all three color channels as follows:

$$E_{key} = \mathrm{mod}\Big((y_R(K) \times y_G(K) \times y_B(K)), 256\Big). \tag{5}$$

Second, E_{key} is used to encrypt the image $IMG(K)$ as follows:

$$IMG_i^e(K) = \mathrm{mod}\Big((y_i(K) + (E_{\mathrm{key}} \times IMG_i^e(K-1) \times 10^{11})), 256\Big). \tag{6}$$

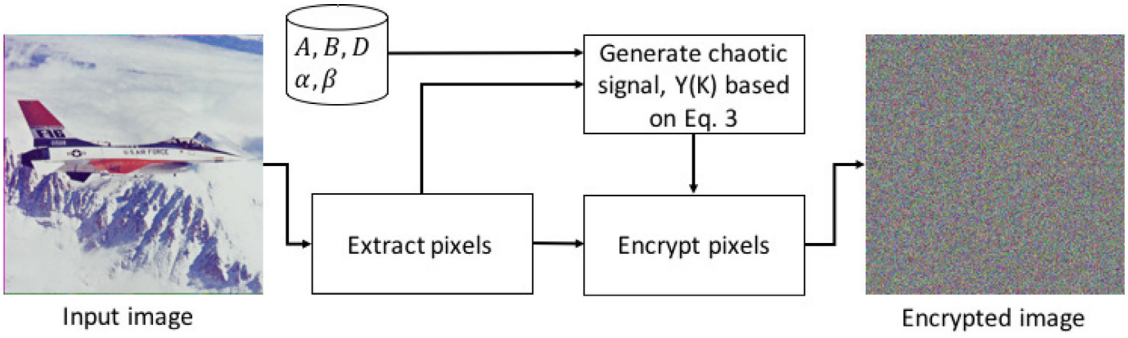


Fig. 1. Flow of processes in the proposed image encryption method.

Finally, we convert the encrypted vector form of $IMG_i^e(K)$ to the matrix form, namely $IMG_i^e(x, y)$, for $x = 1, 2, \ldots, M$, and $y = 1, 2, \ldots, N$.

Here, the expression $IMG_S(K-1)$ is introduced so that the encrypted pixel at position $K-1$ is incorporated to encrypt the next pixel at position K. This is to cater for chosen-plaintext attack so that the proposed encryption method cannot be traced to identify the corresponding output value when given an input values.

Two assumptions are made:

A1. The neuron activation functions $f_j(\cdot)$ are continuously bounded and satisfying

$$l_j^- \le \frac{f_j(u) - f_j(v)}{u - v} \le l_j^+, \tag{7}$$

for any $u,\ v \in \mathbb{R},\ u \neq v,\ j \in \{1, 2, \ldots, n\}$, where l_j^- and l_j^+ are some real value constants, and;

A2. The transmission delay $\tau(t)$ is a time varying delay, and it satisfies $0 \le \tau(t) \le \tau$, where $\tau > 0$ is a constant value.

The flow of processes is illustrated in Fig. 1.

4. Experimental results

The proposed encryption method is implemented in Matlab R2015b running on Windows 7, operating on a personal computer with Intel(R) Core(TM) i7-3630QM CPU @2.40 GHz and 12GB of RAM. For experiment purpose, the reformulated chaotic FCNN system as expressed in (3) is considered with the parameters as follows: $I_i = 0.1, j \in \{1,\ 2,\ 3\}, \tau(t) = 0.07\ |\sin(t)|$ and $\sigma = 0.5$. Letting $f_j(x_j) = \frac{1}{2}\left(|x_j + 1| - |x_j - 1|\right), j \in \{1,\ 2,\ 3\}$, which satisfies the assumption (A_1), we get $l_j^- = -1$, and $l_j^+ = 1$. Specifically,

$$A = \begin{bmatrix} 2.25 & -3.21 & -3.21 \\ -3.2 & 1.1 & -4.39 \\ -3.2 & 4.4 & 0.9 \end{bmatrix}, \tag{8}$$

$$B = \begin{bmatrix} 4.32 & -6.9 & -1.5 \\ -3.01 & 1.2 & -5.01 \\ -3.2 & 4.5 & -2.2 \end{bmatrix}, \tag{9}$$

$$D = \begin{bmatrix} 2.9 & 0 & 0 \\ 0 & 1.001 & 0 \\ 0 & 0 & 1.001 \end{bmatrix}, \tag{10}$$

$$\alpha = \begin{bmatrix} 1/31 & -1/31 & 1/31 \\ 1/31 & 1/31 & -1/31 \\ -1/31 & 1/31 & 1/31 \end{bmatrix}, \tag{11}$$

and

$$\beta = \begin{bmatrix} 1/31 & 1/31 & -1/31 \\ 1/31 & -1/31 & 1/31 \\ -1/31 & 1/31 & 1/31 \end{bmatrix}, \tag{12}$$

with the initial values of the system expressed in Eq. (3) set to $\phi(s) = [-1.5\ \ -0.5\ \ -2]^T$ and $s \in (-\infty, 0]$. All experiments are performed using the LMI toolbox in Matlab using the aforementioned parameter values. On average, the proposed method takes about 27.56 s to encrypt a color image of dimension 512 × 512, which includes the time needed to generate the chaotic signal. Similarly, on average, it takes approximately 27.24 s to decrypt an encrypted image of the same size. The average encryption and decryption time needed for each image in the Uncompressed Color Image Database (UCID) dataset [21] are 13.57 and 13.50 s, respectively, where each image is of dimension 512 × 384. Note that, unless specify otherwise, the result reported under the UCID dataset is the average result for 1338 images. In the following subsections, perceptual semantic masking, information entropy, key space size, key sensitivity test, and statistical attacks are presented, which are all crucial in any image encryption system.

4.1. Masking perceptual semantic

To examine the proposed reformulated FCNN system (3) for successful encryption and decryption, six standard test color images from the USC-SIPI image database [22], namely Baboon, F-16, House, Lake, Lenna and Tiffany, each of dimension 512 × 512 × 3, are considered. The test images are shown in Fig. 2. The trajectory of the reformulated FCNN for Lenna is shown in Fig. 3, which will be utilized to encrypt the test image - Lenna. For completion of discussion, we also conducted the experiments by using the UCID dataset [21], which consists of 1338 images each of dimension 384 × 512.

Fig. 4 illustrate the encrypted images. It is observed that the proposed FCNN based encryption system successfully masked the perceptual semantic of the original image, because the encrypted images appear to be noise.

Next, for each encrypted test image, the entropy of each color channel (i.e., utilized as an image) is considered. Specifically, information entropy gives an indication of randomness and it is defined to express the degree of uncertainties in a system. Let m be the information source, and information entropy H computed is as follows [6]:

$$H = \sum_{i=0}^{N-1} p(m_i) \times \log_2 \frac{1}{p(m_i)} \tag{13}$$

where N represents the total number of gray levels, and $p(m_i)$ is the probability of occurrence of the symbol m_i. The maximum value of H is 8, where all gray levels are equally probable, i.e., uniform distribution. The entropy of the encrypted images obtained

Fig. 2. Original test images.

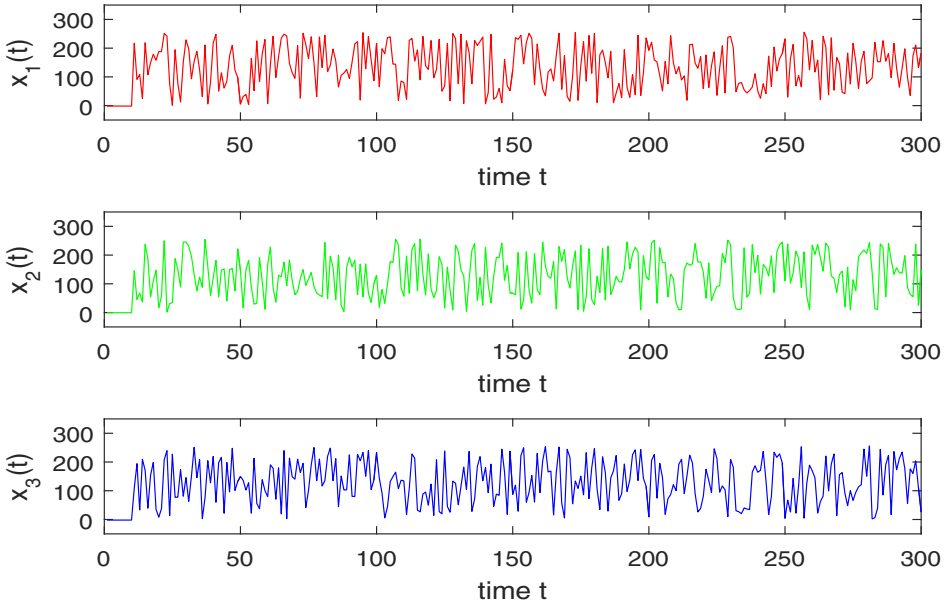


Fig. 3. State trajectories of the reformulated FCNN system (3) with state $x(t)$ for the test image Lenna.

by the proposed method and Hsiao et al.'s method [6] are shown in Table 1. Since only the results of two test images, namely Baboon and Lenna, are reported in Bigdeli et al.'s work [13], their results are omitted from Table 1. Nonetheless, they achieved the entropy of (7.9967, 7.9941, 7.9985) for the RGB channels respectively for Baboon, and similarly (7.9981, 7.9962, 7.9974) for Lenna. It is evidently shown that the entropy value of our proposed method is comparable to those of the conventional methods considered. In addition, the entropy values are high, i.e., near 8 which is the bit-depth of each channel, hence the distributions are close to uniform. It is also noteworthy that both Hsiao et al.'s method [6] and the proposed method are able to increase the entropy of the original test image, which is supported by the result attained by Tiffany - red channel. Specifically, the entropy value was increased from 4.33 to as high as > 7.99 in both methods, which is an increment of about 80%. When considering the UCID dataset [21], the average entropy values achieved are (7.9977, 7.9977, 7.9977) for the RGB channels, respectively, which are near to the maximum value of 8 as expected. Therefore, the proposed method is able to produce encrypted image with high entropy, which is desired in any image encryption algorithm.

The corresponding decrypted images are shown in Fig. 5. The mean square error (MSE) between the original and decrypted images (using the correct keys) are recorded in Table 2 for each of the

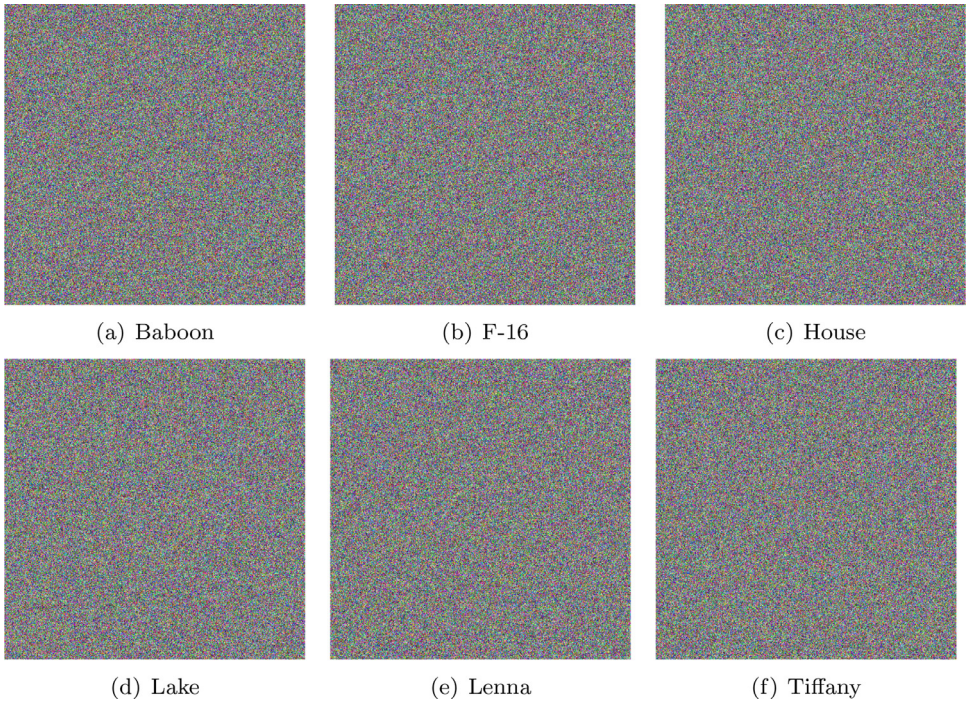


Fig. 4. Encrypted images.

Table 1
Entropy of each color channel for test images encrypted with the proposed method and Hsiao et. al's method [6].

Original	Red	Green	Blue	Encrypted	Red	Green	Blue
Baboon	7.7067	7.4744	7.7522	Ref. [6]	7.9994	7.9993	7.9993
				Fig. 3(a)	7.9982	7.9984	7.9980
F-16	6.7178	6.7990	6.2138	Ref. [6]	7.9994	7.9992	7.9994
				Fig. 3(b)	7.9980	7.9980	7.9978
House	7.4156	7.2298	7.43538	Ref. [6]	–	–	–
				Fig. 3(c)	7.9983	7.9979	7.9980
Lake	7.3124	7.6429	7.2136	Ref. [6]	7.9993	7.9994	7.9994
				Fig. 3(d)	7.9978	7.9976	7.9978
Lenna	7.2531	7.5940	6.9684	Ref. [6]	7.9994	7.9993	7.9993
				Fig. 3(e)	7.9978	7.9976	7.9979
Tiffany	4.3372	6.6643	6.4288	Ref. [6]	7.9993	7.9993	7.9992
				Fig. 3(f)	7.9977	7.9979	7.9978

Table 2
Mean square errors between the original and decrypted color images using the correct key, and PSNR (dB) and SSIM between the original and decrypted gray scale image.

Image	Red	Green	Blue	PSNR	SSIM
Baboon	19.1612	9.5941	15.2595	91.6819	0.9978
F-16	0.0658	63.0157	0.0093	80.0398	0.9999
House	6.1413	133.7544	65.4908	71.2826	0.9982
Lake	0.0254	1.7810	85.3357	68.8005	0.9935
Lenna	11.1867	1.2047	1.2010	106.2036	0.9979
Tiffany	31.6536	92.5257	30.5116	73.1828	0.9998
UCID	234.4820	236.1181	310.0235	85.4939	0.9699

color channels. Note that the MSE is greater than zero because the proposed encryption system involves the manipulation of large integers, which are needed to make the proposed encryption system robust against various attacks. Nonetheless, the decrypted images appear to be perceptually identical to their original counterparts as shown in Fig. 5. For completion of discussion, the corresponding PSNR and SSIM values for all test images are also recorded in the last two columns of Table 2. Here, the PSNR and SSIM values are computed by the using grayscale images, which are generated by using Matlab's function - *rgb2gray*. As expected, the PSNR and SSIM values are high, which agree with Fig. 5 and the MSE values. Similar results are obtained for the UCID dataset [21], and we omit the discussion here.

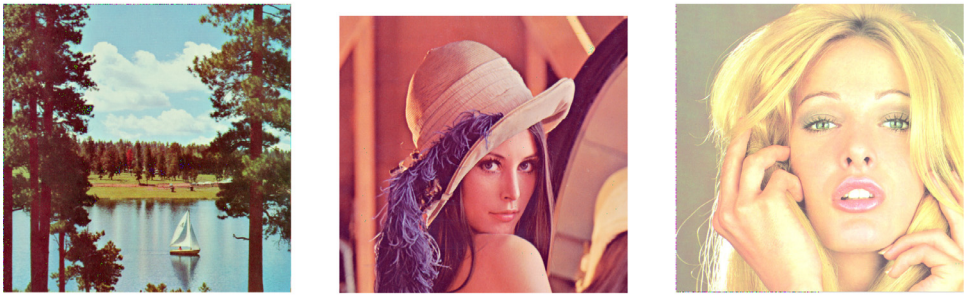
4.2. Secret key space

The brute-force attack commences by considering an one-digit secret key, and then proceed to two-digit secret key, and so on until the maximum length is reached. In order to resist against the brute-force attack, the secret key space should be convincingly large [4]. In this work, the secret key of the proposed algorithm is a combination of three 3×3 parameters that can be expressed as

$$[a_{ij},\ b_{ij},\ d_{ij}], \tag{14}$$

with initial value vector ϕ_i (i.e., 3×1 vector) for $i, j \in \{1, 2, 3\}$. If the precision in question is 10^{-10}, then the key space is $(10^{10})^{30} = 10^{300} > 2^{900}$, which corresponds to a key length greater than 900 bits. This large secret key space of the proposed encryption method is sufficient to ensure that brute-force attack is infeasible. Note that the key space can be further increased accordingly by increasing the precision. A comparison of secret key space

(a) Baboon (b) F-16 (c) Baboon



(d) Lake (e) Lenna (f) Tiffany

Fig. 5. Decrypted images using the correct key.

Table 3
Comparison of secret key space.

Encrypted algorithm	Secret key space
Proposed method	$10^{300} > 2^{900}$
Hsiao & Lee [6]	$10^{196} \approx 2^{651}$
Liu & Wang [23]	$1.0368 \times 10^{114} \approx 2^{379}$
Bigdeli et al. [13]	2^{224}
Liu & Wang [24]	$3.4 \times 10^{94} \approx 2^{314}$

Table 4
Different parameter settings for evaluation on key sensitivity.

Case	Parameter	Original	Mismatched
1	a_{11}	2.25	2.2500000001
2	b_{23}	−5.01	−5.0100000001
3	d_{31}	0.00	0.0000000001
4	ϕ_2	−0.50	−0.4999999999

for the proposed and other conventional encryption methods are summarized in Table 3. It is apparent that the proposed encryption method has the largest key space.

4.3. Key sensitivity test

To illustrate the key sensitivity of the proposed encryption method, the image encrypted by using the parameter settings in (8–12) is decrypted using the same parameters, but with a slight mismatch in the order of 10^{-10} as summarized in Table 4. In other words, the exactly same key is used for decryption, except for the changes summarized in Table 4. Using Lenna as the representative test image, the corresponding decrypted images are shown in Fig. 6. Results suggest that a slight change in any of parameters in the order of 10^{-10} leads to a completely different image (i.e., noise), which do not infer any information about the original image. Therefore, the proposed encryption method is verified to be sensitive to the key in use.

4.4. Statistical attack

To verify the robustness of the proposed encryption method, a statistical analysis is performed for the plaintext and the encrypted images in the following two aspects.

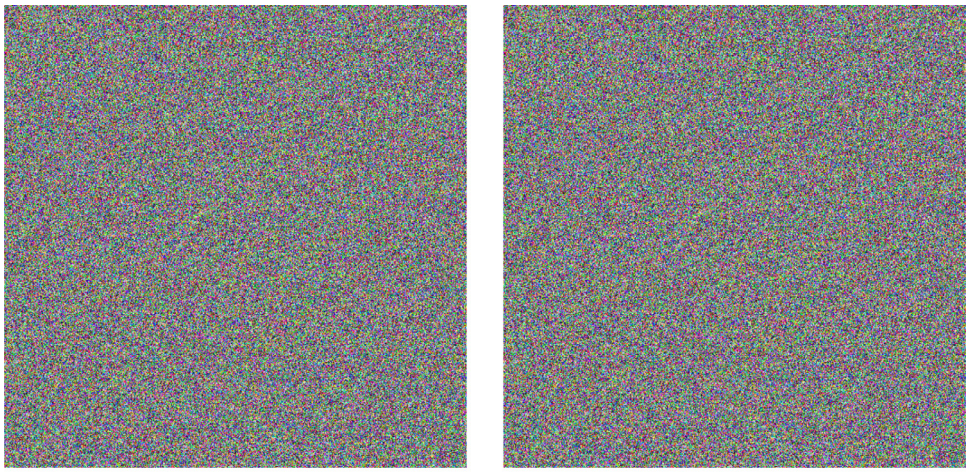
4.4.1. Histogram analysis

A histogram reflects the distribution of the pixel values for a given image. Any effective image encryption method should mask the perceptual meaning of the plaintext image and flatten its histogram (i.e., become near uniform distribution). The histogram before and after encryption are shown in Figs. 7 and 8 for the test images Lenna and Baboon, respectively. It is apparent that the histograms of the encrypted image are fairly uniform and hence significantly different from that of the original images. The outcome for other images are similar and we omit the results here. All in all, the histogram does not provide any useful clue for an attacker to launch any statistical attacks on the proposed image encryption procedure. Therefore, the proposed encryption method successfully randomized the pixels.

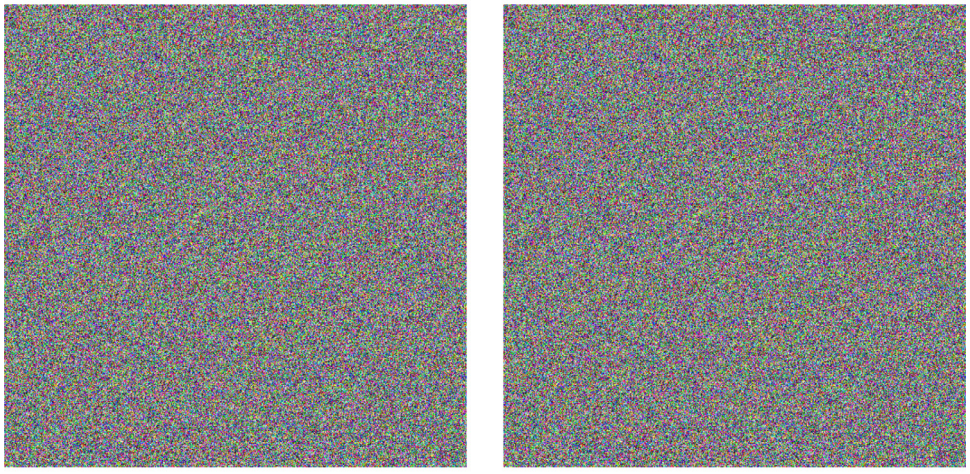
4.4.2. Correlation analysis

In the plain image, the correlation coefficients are high in general, because the adjacent pixels in all three directions, i.e., the horizontal, vertical, and diagonal directions, often have the same or a similar pixel values. However, in the encrypted image, the correlation coefficients should be low (e.g., close to zero), i.e., uncorrelated. To illustrate the pixel correlations before and after encryption, 2000 pixels are randomly selected as the x value. Correspondingly, one of the 8 neighbors of each selected pixel is chosen to be its y value. As the representative examples, the graph of these points $P(x, y)$ are plotted in Figs. 9 and 10 for the test images Lenna and Baboon, respectively. As expected, the highly correlated pixels (i.e., before encryption) show little to no correlation after encryption.

For further analysis, the correlation coefficients of the plain image and the encrypted image are computed as follows:

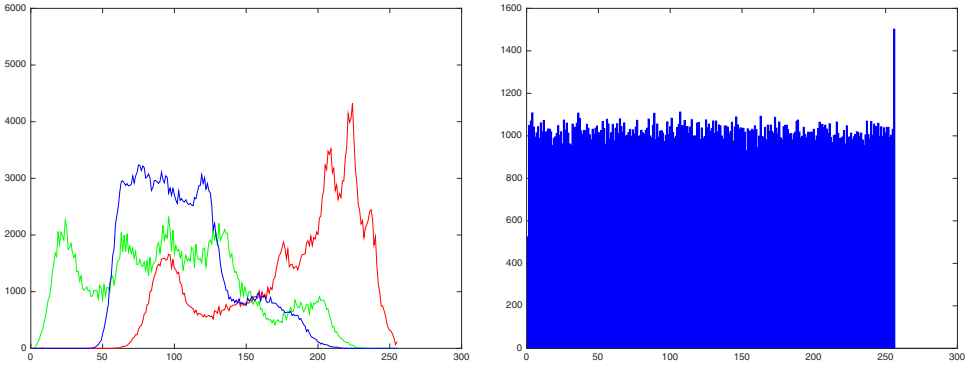


(a) $a_{12} = 2.2500000001$ (b) $b_{23} = -5.0100000001$



(c) $d_{31} = 0.0000000001$ (d) $\phi = -0.4999999999$

Fig. 6. Images decrypted from Fig. 4(e) by using the wrong keys summarized in Table 4.



(a) Histogram of (a) (b) Histogram of (b)

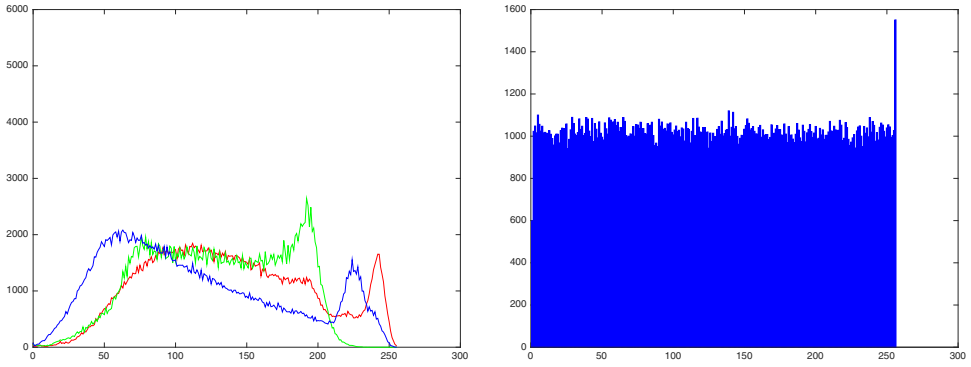
Fig. 7. Histogram of the original and encrypted Lenna image.

1. For each of the RGB channels in a color image, randomly select 2000 pairs of two adjacent pixels in the horizontal, vertical, or diagonal direction. Altogether there are 6000 points.
2. The correlation coefficient, denoted by r_{xy}, is calculated by using the following formulae [25]

$$r_{xy} = \frac{|cov(x,y)|}{\sqrt{D(x)}\sqrt{D(y)}}, \tag{15}$$

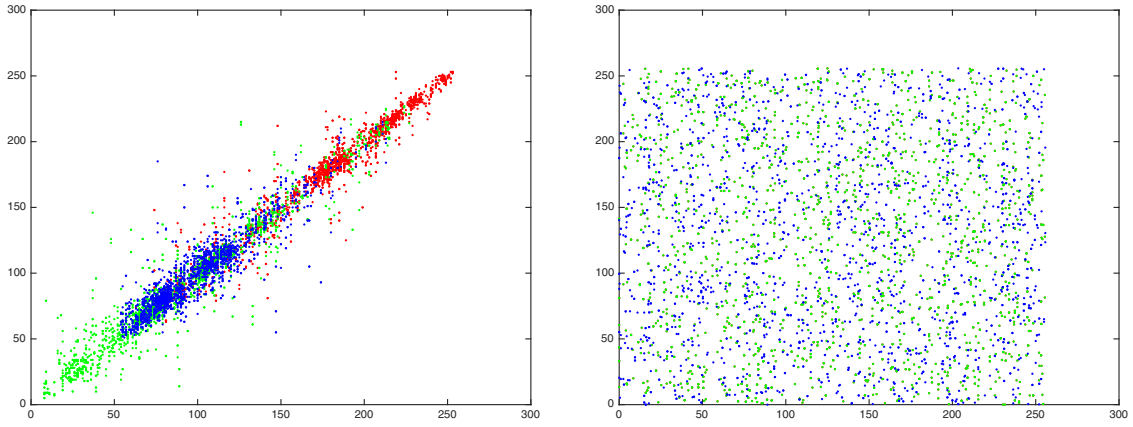
$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2,\ D(y) = \frac{1}{N}\sum_{i=1}^{N}(y_i - E(y))^2, \tag{16}$$

$$cov(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)), \tag{17}$$



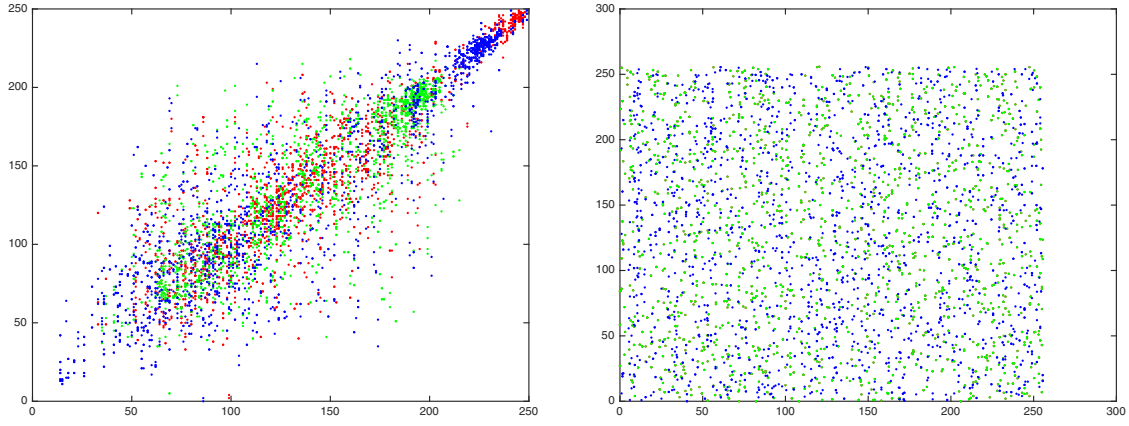
(a) Histogram of (a) (b) Histogram of (b)

Fig. 8. Histogram of the original and encrypted Baboon image.



(a) Pixel correlation of (a) (b) Pixel correlation of (b)

Fig. 9. Graph of pixel correlation before and after encryption - Lenna.



(a) Pixel correlation of (a) (b) Pixel correlation of (b)

Fig. 10. Graph of pixel correlation before and after encryption - Baboon.

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i,\ \ E(y) = \frac{1}{N}\sum_{i=1}^{N} y_i, \tag{18}$$

where x_i and y_i denote pixel values of two different pixels in the image itself or between two different images. Here, $cov(x, y)$ indicates the covariance, $D(x)$ and $D(y)$ are the variance, and $E(x)$ and $E(y)$ are the means.

The results are recorded in Table 5, which suggest that the proposed encryption method can effectively randomize the pixel values because the correlation coefficients is small in magnitude for all color channels. The encrypted pixel values show little to no correlation, which also agree with the graphs in Figs. 9 and 10 for the case of Lenna and Baboon, respectively. The rest of the images (including those from the UCID dataset [21]) behave similarly and we omit the results here.

The proposed method is also compared with the conventional encryption methods in terms of correlation coefficients, and the results are recorded in Table 6. Results suggest that, for all encryption methods considered, the neighboring pixels in the encrypted

Table 5
Correlation coefficients of the red, green, and blue channels of the original and encrypted color images ($\times 10^{-2}$).

	Red	Green	Blue
Plaintext image			
Baboon	88.2132	81.5407	90.2738
F-16	94.3991	94.7277	93.9894
House	91.6448	90.0456	94.1567
Lake	96.3884	97.2590	96.5515
Lenna	97.0725	95.1062	90.0020
Tiffany	94.7014	91.9219	93.6470
Average	93.7366	91.7669	93.1034
UCID	93.7197	94.0391	93.8108
Encrypted image			
Baboon	−1.8201	1.0509	−0.030352
F-16	−0.3055	0.7109	2.5270
House	0.7876	−0.8849	2.2173
Lake	−3.3869	−2.4822	0.9267
Lenna	−0.8916	−3.4630	−2.5978
Tiffany	1.2566	1.9780	1.9783
Average	−0.7267	−0.5151	0.3361
UCID	−0.0407	−0.0239	0.0345

Table 6
Comparison of correlation coefficients of the red, green, and blue channels among the proposed and conventional methods ($\times 10^{-2}$).

Method	Red	Green	Blue
Proposed method (Table 5)	−0.7267	−0.5151	0.3361
Hsiao & Lee [6]	1.6231	1.3235	3.0774
Wang et al. [7]	−1.0889	−1.8110	−0.6104
Hussain et al. [26]	−2.0889	−2.8110	−0.5104

image are of low correlation. It is also observed that the performance of the proposed method is comparable to those of the conventional methods considered.

4.5. Chosen-plaintext attack

One of the significant sensitivity analysis is plaintext sensitivity, which is also known as chosen-plaintext attack. Plaintext sensitive means that any tiny disturbance of the plain-image will lead to dramatic changes in the encrypted image. Therefore, in order to measure the difference between the resulting encrypted images, there are two criteria, namely, number of pixels change rate (NPCR), which is defined as:

$$\mathrm{NPCR}_{R,G,B} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} D_{R,G,B}(i,j)}{M \times N} \times 100\%, \tag{19}$$

and unified average changing intensity (UACI), which is expressed as:

$$\mathrm{UACI}_{R,G,B} = \sum_{i=1}^{M}\sum_{j=1}^{N} \frac{|C_{R,G,B}(i,j) - C'_{R,G,B}(i,j)|}{255 \times M \times N} \times 100\%, \tag{20}$$

subject to

$$D_{R,G,B}(i,j) = \begin{cases} 0, & \text{if } C_{R,G,B}(i,j) = C'_{R,G,B}(i,j); \\ 1, & \text{otherwise.} \end{cases}$$

Here, $C_{R,\,G,\,B}(i,j)$ and $C'_{R,G,B}(i,j)$ denote the two different encrypted images in which their plaintext images $P_{R,\,G,\,B}(i,j)$ and $P'_{R,G,B}(i,j)$ differ only by one pixel. Note that higher NPCR value implies better performance. On the contrary, lower value of UACI implies better performance. The NPCR and UACI results for the test images are recorded in Table 7. These results suggest that the proposed encryption FCNNs-based method is robust against chosen-plaintext attack. For example, the average NPCR is 99.95% even when the input images differ only by 1 pixel.

Table 7
NPCR and UACI for two encrypted images where their corresponding plaintext images differ only by one pixel. The encrypted images are generated by using the same secret key.

	NPCR		
Component	Red	Green	Blue
Baboon	0.9991	0.9990	0.9990
F-16	0.9999	0.9998	0.9999
House	0.9993	0.9993	0.9993
Lake	1.0000	1.0000	1.0000
Lenna	0.9991	0.9992	0.9991
Tiffany	0.9999	1.0000	1.0000
Average	0.9995	0.9995	0.9995
UCID	0.9961	0.9961	0.9961
	UACI		
Baboon	0.3344	0.3337	0.3330
F-16	0.3327	0.3328	0.3329
House	0.3333	0.3334	0.3336
Lake	0.3341	0.3337	0.3332
Lenna	0.3335	0.3334	0.3330
Tiffany	0.3339	0.3333	0.3333
Average	0.3336	0.3334	0.3332
UCID	0.3346	0.3346	0.3346

Table 8
Average value for NPCR and UACI.

Image	NPCR (mean)			
	Color Channel	Red	Green	Blue
Plaintext	Proposed method	0.9995	0.9995	0.9995
	Hsiao & Lee [6]	0.9961	0.9961	0.9961
	Liu & Wang [23]	0.9961	0.9962	0.9958
	UACI (mean)			
Encrypted	Proposed method	0.3339	0.3333	0.3330
	Hsiao & Lee [6]	0.3345	0.3349	0.3344
	Liu & Wang [23]	0.3355	0.3341	0.3335

For completion of discussion, the NPCR and UACI scores attained by the proposed encryption method are also compared with those of the conventional methods. It is observed that all considered methods exhibit high NPCR and low UACI, suggesting their high plaintext sensitivity property. Nonetheless, it is observed that the proposed method outperforms the conventional encryption methods considered in both NPCR and UACI (Table 8).

5. Conclusions

In this work, we reformulated the FCNN to encrypt color image. Specifically, we use each generated FCNNs chaotic signal expressed in (1) together with each pixel of the original image to generate the output encrypted pixel, which makes the proposed encryption system robust against various cryptanalysis, including plaintext-only and chosen-plaintext attacks. To add another layer of security, the chaotic signal can begin at any point in the originally generated chaotic signal. The key sensitivity was verified to be in the order of 10^{-10}. In addition, plaintext sensitivity tests and statistical analysis against the conventional methods demonstrated the effectiveness of our proposed method.

As future work, fast computation method for the proposed FCNN will be designed to meet the requirements of real-time applications. In addition, the proposed FCNN will be further refined for applications in other domains, including audio and video where the samples are highly correlated.

Acknowledgment

This work was supported by the Fundamental Research Grant Scheme (FRGS) MoHE Grant (FP051-2016) and the University of Malaya HIR under Project UM.C/625/1/HIR/MOHE/ENG/42.

References

[1] T. Stutz, A. Uhl, A survey of h.264 AVC/SVC encryption., IEEE Trans. Circuits Syst. Video Techn. 22 (3) (2012) 325–339.
[2] A. Massoudi, F. Lefebvre, C. De Vleeschouwer, B. Macq, J.-J. Quisquater, Overview on selective encryption of image and video: challenges and perspectives, EURASIP J. Inf. Secur. 2008 (2008) 5:1–5:18, doi:10.1155/2008/179290.
[3] K. Ho, 41 up-to-date facebook facts and stats, 2015, (http://blog.wishpond.com/post/115675435109/40-up-to-date-facebook-facts-and-stats).
[4] R. Enayatifar, H.J. Sadaei, A.H. Abdullah, M. Lee, I.F. Isnin, A novel chaotic based image encryption using a hybrid model of deoxyribonucleic acid and cellular automata, Opt. Lasers Eng. 71 (2015) 33–41.
[5] G. Zhou, D. Zhang, Y. Liu, Y. Yuan, Q. Liu, A novel image encryption algorithm based on chaos and line map, Neurocomputing 169 (2015) 150–157.
[6] H.-I. Hsiao, J. Lee, Color image encryption using chaotic nonlinear adaptive filter, Signal Process. 117 (2015) 281–309.
[7] X. Wang, L. Teng, X. Qin, A novel colour image encryption algorithm based on chaos, Signal Process. 92 (4) (2012) 1101–1108.
[8] S. El Assad, M. Farajallah, A new chaos-based image encryption system, Signal Process. 41 (2016) 144–157.
[9] J. Zhao, S. Wang, Y. Chang, X. Li, A novel image encryption scheme based on an improper fractional-order chaotic system, Nonlinear Dyn. 80 (4) (2015) 1721–1729.
[10] C. Li, Cracking a hierarchical chaotic image encryption algorithm based on permutation, Signal Process. 118 (2016) 203–210.
[11] S. Lian, A block cipher based on chaotic neural networks, Neurocomputing 72 (4) (2009) 1296–1301.
[12] S. Wen, Z. Zeng, T. Huang, Q. Meng, W. Yao, Lag synchronization of switched neural networks via neural activation function and applications in image encryption, IEEE Trans. Neural Netw. Learn. Syst. 26 (7) (2015) 1493–1502.
[13] N. Bigdeli, Y. Farid, K. Afshar, A novel image encryption/decryption scheme based on chaotic neural networks, Eng. Appl. Artif. Intell. 25 (4) (2012) 753–765.
[14] T.A. Fadil, S.N. Yaakob, B. Ahmad, A hybrid chaos and neural network cipher encryption algorithm for compressed video signal transmission over wireless channel, in: Electronic Design (ICED), 2nd International Conference on, IEEE, 2014, pp. 64–68.
[15] S. Chatzidakis, P. Forsberg, L.H. Tsoukalas, Chaotic neural networks for intelligent signal encryption, in: Information, Intelligence, Systems and Applications, IISA, The 5th International Conference on, IEEE, 2014, pp. 100–105.
[16] L.O. Chua, L. Yang, Cellular neural networks: theory, IEEE Transa. Circuits Syst. 35 (10) (1988a) 1257–1272.
[17] L.O. Chua, L. Yang, Cellular neural networks: applications, IEEE Trans. Circuits Syst. 35 (10) (1988b) 1273–1290.
[18] T. Yang, L.-B. Yang, C.W. Wu, L.O. Chua, Fuzzy cellular neural networks: applications, in: Cellular Neural Networks and their Applications, 1996. CNNA-96. Proceedings., 1996 Fourth IEEE International Workshop on, IEEE, 1996, pp. 225–230.
[19] T. Yang, L.-B. Yang, C.W. Wu, L.O. Chua, Fuzzy cellular neural networks: theory, in: Cellular Neural Networks and their Applications. CNNA-96. Proceedings., Fourth IEEE International Workshop on, IEEE, 1996, pp. 181–186.
[20] A. Abdurahman, H. Jiang, Z. Teng, Finite-time synchronization for fuzzy cellular neural networks with time-varying delays, Fuzzy Sets Syst. 297 (2016) 96–111.
[21] G. Schaefer, M. Stich, UCID - an uncompressed colour image database, in: In Storage and Retrieval Methods and Applications for Multimedia 2004, Proceedings of SPIE, 5307, 2004, pp. 472–480.
[22] The usc-sipi image database, (http://sipi.usc.edu/database/database.php).
[23] H. Liu, X. Wang, Color image encryption using spatial bit-level permutation and high-dimension chaotic system, Opt. Commun. 284 (16) (2011) 3895–3903.
[24] H. Liu, X. Wang, Triple-image encryption scheme based on one-time key stream generated by chaos and plain images, J. Syst. Softw. 86 (3) (2013) 826–834.
[25] R. Rhouma, S. Meherzi, S. Belghith, OCML-based colour image encryption, Chaos, Solitons & Fractals 40 (1) (2009) 309–318.
[26] I. Hussain, T. Shah, M.A. Gondal, An efficient image encryption algorithm based on s 8 s-box transformation and nca map, Opt. Commun. 285 (24) (2012) 4887–4890.