Hindawi
Scientific Programming
Volume 2022, Article ID 2628885, 17 pages
https://doi.org/10.1155/2022/2628885

Research Article

An Integration of New Digital Image Scrambling Technique on PCA-Based Face Recognition System

Eimad Abusham,[1] **Basil Ibrahim**,[1] **Kashif Zia**,[2] **and Sanad Al Maskari**[1]

[1]*Sohar University, Faculty of Computing and Information Technology, Sohar, Oman*
[2]*University of Glasgow, Glasgow, UK*

Correspondence should be addressed to Basil Ibrahim; basilshakkak@gmail.com

Received 22 July 2022; Revised 17 October 2022; Accepted 10 November 2022; Published 25 November 2022

Academic Editor: Jianping Gou

Systems using biometric authentication offer greater security than traditional textual and graphical password-based systems for granting access to information systems. Although biometric-based authentication has its benefits, it can be vulnerable to spoofing attacks. Those vulnerabilities are inherent to any biometric-based subsystem, including face recognition systems. The problem of spoofing attacks on face recognition systems is addressed here by integrating a newly developed image encryption model onto the principal component pipeline. A new model of image encryption is based on a cellular automaton and Gray Code. By encrypting the entire ORL faces dataset, the image encryption model is integrated into the face recognition system's authentication pipeline. In order for the system to grant authenticity, input face images must be encrypted with the correct key before being classified, since the entire feature database is encrypted with the same key. The face recognition model correctly identified test encrypted faces from an encrypted features database with 92.5% accuracy. A sample of randomly chosen samples from the ORL dataset was used to test the encryption performance. Results showed that encryption and the original ORL faces have different histograms and weak correlations. On the tested encrypted ORL face images, NPCR values exceeded 99%, MAE minimum scores were over (>40), and GDD values exceeded (0.92). Key space is determined by $u(2^{size(A_0)})$ where A_0 represents the original scrambling lattice size, and u is determined by the variables on the encryption key. In addition, a NPCR test was performed between images encrypted with slightly different keys to test key sensitivity. The values of the NPCR were all above 96% in all cases.

1. Introduction

Face recognition systems identify human faces and can differentiate between them by processing and storing visual patterns in visual data [1]. A facial recognition system provides users with a number of advantages, including passive authentication [2], by which authenticity can be established simply by being present. Video surveillance, access control, forensics, and social media are some of the security-related applications of facial recognition [3–11]. In accordance with [12], facial recognition systems follow the following stages: A preprocessing stage is first performed on the data. Aligning the area of interest after detecting faces in visual input. Using the preprocessed input, face features are extracted in a second step. To determine whether a face matches a database of features, the features of the face are compared. It is possible to verify a specific target or identify a facial feature based on matching results. Figure 1 shows a diagram of the facial recognition process.

In terms of biometric authentication, face recognition shares many advantages and disadvantages with other biometric methods. Biometric authentication generally provides greater security than traditional passwords [13]. Every individual has unique biometric traits, which make biometric forgeries difficult [14], and it also prevents false authentication because the registered person must be present to verify authenticity [15]. For example, a biometric authentication system could be used to protect the integrity of results obtained by studies that involve analysis of medical patterns obtained from samples taken from a predetermined

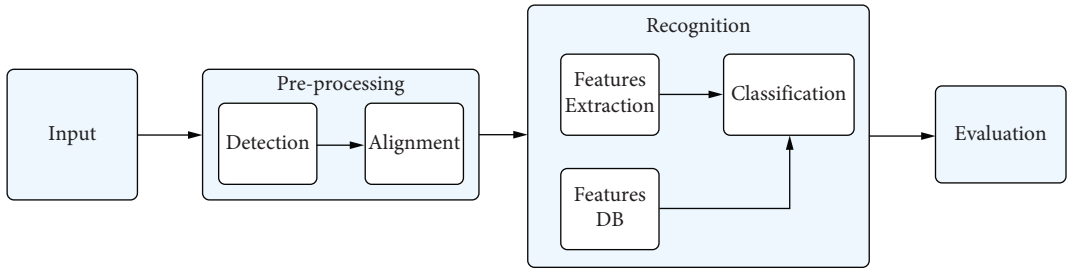

Figure 1: Stages of the facial recognition system [12].

set of subjects such as in the works of [16, 17]. Subjects involved in such studies can be identified and verified with biometric authentication systems before proceeding with subsequent medical analysis procedures thereby ensuring that the subject does belong to the main group under study or any subset of that group that requires special procedures. The disadvantage of biometric authentication systems is that they are vulnerable to attacks involving deep learning and machine learning models that spoof the biometrics [13].

False testimony about biometrics is submitted by attackers to gain authenticity [18]. In addition to artificial synthesis and reply attacks, there are also wolf attacks [13] and replay attacks [19]. In addition to a number of adversarial attacks [13] and poisoning attacks [20], machine learning and deep learning models are also susceptible to poisoning attacks.

A method is presented in this paper for preventing spoofing attacks on facial recognition systems by integrating an image encryption model into the process. To train and test a face recognition model based on principle component analysis (PCA), preprocessed face images are encrypted using an image encryption model. A feature extracted from an input face image has to match the key used to encrypt the image in order to be correctly identified or verified. The extracted features enrolled in the features database are encrypted. Attackers must encrypt face images with the correct key in addition to copying and submitting authenticated individuals' images to a system to minimize the effectiveness of spoofing attacks.

In addition to offering high encryption performance and resistance to brute-force attacks, the image encryption model provides an added layer of security. An image encryption model is developed that is based on outer totalistic cellular automatas (OTCAs) and gray code for use in recognition processes. Pixels are replaced with Gray Code, while images are scrambled using CA. Using mathematical operations, a pixel's substitution changes its values, then reverses those operations to return its values [21]. As part of the process of scrambling images, pixels are moved on the image in order to break the high correlation between adjacent pixels [22]. CA is an excellent choice for image scrambling applications [23], since it can generate complex structures from simple structures. Following is the order in which the remaining sections of this paper are presented. Second, the study reveals its work in image scrambling; third, it explores the methodology it used; fourth, it summarizes the results; and last, it presents its conclusions.

2. Related Work

CAs consist of an infinite array of discretely updating cells, with each cell changing its state according to a universal rule depending on its present state and the states of its neighbors. Based on [24], CA-based image encryption uses direct operations on the pixels of the image to encrypt the data. There are several advantages to CA image encryption, including its ease of implementation and high security [24]. As CA-based image scrambling methods [25–30] were developed, according to these methods, CA scrambling can break highly correlated pixels while remaining high-performing and resistant to a variety of attacks.

A proposal in [26] uses CA for watermarking and scrambling. CA rules are studied using fractal box dimensions in order to determine chaotic characteristics. After the image has been scrambled with a specific lattice and its evolution over a certain number of generations, it is scrambled according to the selected CA. This method of scrambling images was first used for watermarking. Using this scheme, watermarked images are more resistant to attacks such as noise, cropping, and JPEG compression. A gray image encryption scheme was developed by [30] using 2D CA. A binary image represents a bit, and eight images are created. With the B3/S1234 CA rule, eight binary images were generated as an initial configuration lattice. As 8 binary lattices are developed based on 8 binary images of the original image, both the value and position of pixels are simultaneously changed.

A study by [27] investigated how other 2D-OTCA rules would perform when scrambling images besides Game of Life. By using Von Neumann neighborhood configuration instead of Moore's law, the authors reduce rules space and computation time. A number of generations and boundary conditions are used to evaluate scrambling performance using GDD (gray difference degree) OCTA rules. As a part of the proposed method, a random lattice is generated and evolved k times over the course of time. A matrix is used to scramble the subject image based on the evolving lattice. The initial lattice is empty. Then, starting from the top-leftmost cell, the locations of the pixels in the original subject image are used to identify the active cells. When pixels corresponding to dead cells are scrambled, rows major are copied to pixels corresponding to dead cells. This technique achieved the highest GDD on Rule 171 when compared to other proposed techniques. In addition, the technique was much faster than the other methods when it came to computation time.

Image scrambling was achieved using 2D CA in [25]. In their study, the authors looked at different configurations, such as the number of evolved generations, neighborhood configuration, boundary conditions, and rules with lambda values close to critical values, when scrambling performance was determined. A lattice from which an image is scrambled is developed using all the lattices obtained from the initial lattice. The lattice is initially empty, and then pixel values from the original subject image are multiplied by the pixel locations corresponding to the live cells in the first scrambling matrix, starting with the top-leftmost cell. The pixels in locations that have already been filled are skipped after the matrices have been scrambled. Dead pixels on the scrambling matrix are copied to corresponding pixel locations on the original image in row-major order. Compared to scrambling with fewer generations, scrambling with more generations results in a better GDD. The Moore configuration with periodic boundary conditions could also be used to increase GDD. The lambda values tested ranged from 0.20703 to 0.41404. According to the image tests, Rule 224: Game of Life achieved the highest GDD value.

A 2D CA image scrambling technique proposed by [25] was modified by [28] to achieve better GDD scrambling. Similarly, scrambling occurs in all evolutionarily evolved lattices. A row-major lattice is constructed from the pixels in the original image that correspond to the live cells in the scrambling matrix. Additionally, the remaining pixels are copied to the remaining row-major locations. The same procedure is repeated if there are more scrambling matrices. Scrambled lattices are created by applying Game of Life 224 rules. To improve GDD, periodic boundary conditions, Moore's neighborhood, and eight generations are used. It was the combination of periodic boundary conditions, Moore's neighborhood, and eight generations that produced the highest GDD of 0.9551.

According to [31], images can be scrambled with ECA. In this study, ECA rules were used to test scrambling performance on classes 3 and 4 rules. Scrambling was used to convert the original images into 1D vectors. During the scrambling process, a 1D lattice generated at random is evolved for k generations. A 1D lattice corresponding to the locations of live cells on the scrambling lattice is created by copying pixels from the original image. In the same way, the remaining scramble matrices that have previously been filled are skipped. After the original image is scrambled, the remaining pixels are copied to dead cells. Matrixes are then created by transforming the 1D vector into 2D. When combined with ECA scrambling, GDD can be just as effective as 2D CA and, in some cases, may even be more effective. As a result of combining Rule 22 with boundary conditions and ten generations, the GDD was high. Compared to the tested ECA rules for class 3 (22, 30, 126, 150, and 182), class 4 rule 110 achieved a higher GDD.

In [29], an image encryption method based on 2D OTCA is proposed. When rules 534 and 816 are applied to the original image, pixels values and locations are simultaneously changed. The method's robustness is demonstrated by histograms and entropies. There is a large key space and a high degree of sensitivity. There was a NPCR of about 100%, an entropy of more than 7.2, and a correlation of almost zero on all test images. It is not possible to identify encrypted images from original images using histogram analysis.

There are many linear techniques for face recognition, but PCA is one of the most widely adopted [32]. According to [33], PCA is used to recognize faces. A PCA transforms data linearly into a new coordinate corresponding to the maximum variance direction [34]. PCA's application to face recognition is described in this document [32]. Face images can be modeled using PCA to extract features, creating eigen faces based on eigenvectors. Eigen values with high values and their corresponding eigen vectors are determined from face data vectors using the covariance matrix.

Using PCA and ANFIS (adaptive neuro-fuzzy inference system), an efficient pose-invariant face recognition system was developed in [35]. An ANFIS classifier is used to recognize images from a variety of poses using PCA as a feature extractor. Data sets of training images of faces are scored by PCA algorithms to enable classification. Correct recognition rates are greatly improved by using ANFIS.

Based on PCA and logistic regression, [36] proposed a face recognition system. An experimentation dataset is used in the face recognition pipeline to reduce the dimensions of features. A logistic regression classifier was proposed for accurate face recognition. A two-dataset analysis was conducted to determine the classification's efficiency.

An automatic attendance system was created by [37] to track and record the attendance of individuals within an organization. By eliminating the need to manually take attendance, automated attendance systems enable organizations to optimize their processes. In order to implement the system, a face recognition system is used. As part of the system, Haar cascades are used to detect faces. In order to test the system's capability to recognize faces, PCA and LDA were applied to the Olivetti dataset.

Image encryption cannot yet be integrated into face recognition processes because there are not enough studies in this area. A great deal of research is being conducted on improving the accuracy of face recognition or deploying it effectively in a wide variety of applications. This work integrates OTCA and Gray Code facial recognition algorithms with a new image encryption scheme. Therefore, the recognition system can correctly identify encrypted faces as a result.

3. Methodology

A description of the methods and configurations used in the image encryption scheme is presented in this section. In the following step, we demonstrate a short PCA face recognition algorithm as well as integrate an image encryption scheme into the recognition process.

3.1. Image Encryption Scheme. In this process, gray code-based pixels are substituted, followed by 2D OTCA-based pixels scrambling. Pixel values are replaced with Gray Code representations in the pixel substitution process. A random lattice is generated to scramble pixels using the OTCA Conway's Game of Life rule.

3.1.1. Gray Code Pixels Substitution. This phase involves replacing the Gray Code integer corresponding to a given pixel at coordinates (i, j). XOR operations are applied on adjacent binary bits in the binary representation of integers to generate Gray Code, which is then appended to the left of the string that contains the first bit. Images can be formatted with different bit depth. This bit depth represents the number of bits contained in each pixel; therefore, bit depth is the length of the binary string n used to represent pixels. The higher the value of an image's bit depth (or length of binary string n) is, the more space (file size) is required to store the image. Note that bit depth does not affect an image's resolution. (Algorithm 1) is applied on binary pixels values of images to convert them to their Gray Code representations

To perform pixel substitution, the original image is processed to replace any pixels at positions (i, j) by equivalent gray codes corresponding to the corresponding gray levels. Thus $I'(i, j) = integer[GrayCode(I(i, j))]$.

3.1.2. Image Scrambling with CA. After Gray Code pixels substitution in phase one, the image is scrambled using an evolved 2D-OTCA lattice. According to the method used, the type of 2D CA used is outer totalistic cellular automata (2D OTCA). OTCA rules work by updating cells based on the current state of a cell and its neighboring cells, as described in [23]. As a result of OCTA rules, it is possible to describe the new state of cells using a transition function v as follows:

$$I^{t+1}(i, j) = v\left(I^t(i, j), \sum_{i', j'} I^t(i', j')\right), \quad (1)$$

where $I^t(i', j')$ are cells in $I^t(i, j)$ neighborhood.

(1) Neighborhood Configuration. Once the OCTA transition function is defined, neighborhood configuration needs to be specified for any cell in 2D lattice. Der von Neumann neighborhood scheme and Moore neighborhood scheme [38] are popular neighborhood schemes depicted in Figure 2. In Von Neumann's neighborhood (N_{vN}) a cell at coordinates (i', j') is a neighbor to cell (i, j) if it is an adjacent cell on one of the four directions north, east, west, or south to central cell (i, j). Range of cell's neighborhood can be extended given a radius r, with that a cell at coordinates (i', j') is a neighbor to cell (i, j) at radius r if it satisfies the following rule:

$$(i', j') \in N_{vN}(i, j, r) if |i' - i| + |j' - j| \leq r. \quad (2)$$

As for Moor's neighborhood (N_M) a cell at coordinates (i', j') is a neighbor to cell (i, j) if it is an adjacent cell on one of the four directions north, east, west, or south or on a diagonal direction as well as to the central cell (i, j). Similarly range of neighborhood for a cell at coordinates (i, j) can be extended for a given radius r. With that, a cell at coordinates (i', j') is a neighbor to cell (i, j) at radius r if it satisfies the following rule:

$$(i', j') \in N_M(i, j, r) if |i' - i| \leq r and |j' - j| \leq r. \quad (3)$$

(2) Boundary Conditions. As 2D lattices with CA rules are finite, neighboring cells at lattice bounds should be specified. A closed boundary condition (CBC) or a periodic boundary condition (PBC) can be applied to cells at extremes [40]. In CBC, adjacent cells considered naught are those adjacent to the extreme cells [41]. As for the PBCs, the cells at the extremes become adjacent to one another, so in a 2D rectangular lattice the leftmost cells are next to the rightmost cells, the top row cells are next to the bottom row, and the cells on corners are also adjacent, thus forming a toroid shape lattice [25].

(3) Conway's Game of Life. Conway's Game of Life (CGL) is the most famous universal automaton [42] ever invented by John Conway in 1970. OTCA describes a cell's state based on its current status and eight nearby cells. CGL applies Moor's Neighborhood configuration to its OCTA transition function. According to [43] in the CA GoL (Game of Life) rules (CGL and other discovered GoL rules) neighboring cells are cells that are directly touching a candidate cell. Therefore in CGL neighborhood configuration is strictly confined to Moor's neighborhood configuration in square grid. There are other investigated shapes of grids such as hexagonal, pentagonal, and triangular grids, that may or may not have their own set of discovered GoL rules [44], but CGL does not satisfy the requirements to be a GoL rule in such grids. In CGL, cells can either be alive or dead based on their state and the states of neighboring cells. In CGL, the transition of cells between available states is governed by the following rules (see Figure 3 for an example):

(i) For a cell at coordinates (i, j) such that $I^t(i, j) = 0$ if $\sum_{i', j'} I^t(i', j') = 3$ for $(i', j') \in N_M(i, j, r)$ then $I^{t+1}(i, j) = 1$ otherwise $I^{t+1}(i, j) = 0$.

(ii) For a cell at coordinates (i, j) such that $I^t(i, j) = 1$ if $2 \leq \sum_{i', j'} I^t(i', j') \leq 3$ for $(i', j') \in N_M(i, j, r)$ then $I^{t+1}(i, j) = 1$ otherwise $I^{t+1}(i, j) = 0$.

3.1.3. Scrambling Algorithm. Using the proposed scrambling algorithm, the original image is first transformed with Gray Code, then it is scrambled using an evolved 2D lattice with CGL OTCA rules on a network with PBC. The following sections describe the steps in encrypting and decrypting data.

(1) Encryption Process

(1) Convert original image to its grayscale version, and then transform all pixels in I to its corresponding Gray Code integer equivalent in I'. That is value of pixel at coordinates (i, j) in I' can be determined by:

$$I'(i, j) = integer[GrayCode(I(i, j))]. \quad (4)$$

(2) Generate random lattice A_0 with exactly the same width and height as I. Values of lattice pixels can either be 1 (alive) or 0 (dead).

(3) Apply CGL OTCA transition function v_{CGL} on A_0 with N_M and PBC for k generations yielding A_k.

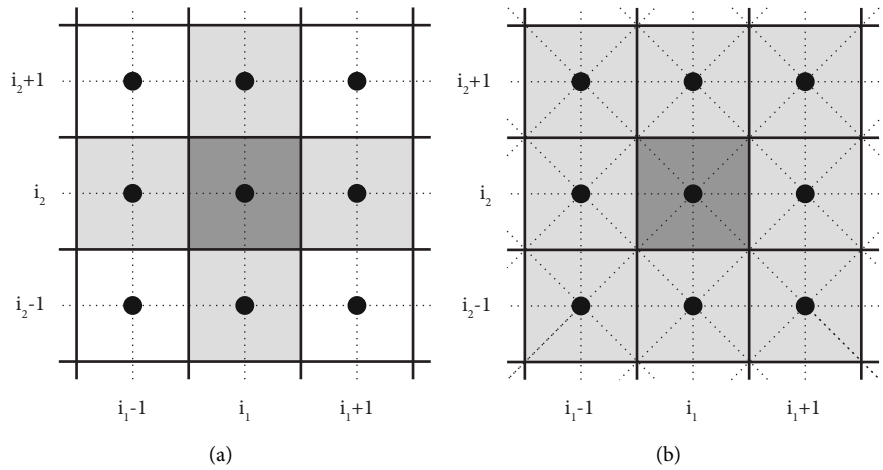

Figure 2: Neighborhood configurations. (a) Van Neumann's neighborhood at $r = 1$. (b) Moore's neighborhood at $r = 1$ [39].

(4) Combine A_k and $A_n\,(0 < n < k)$ on an initially empty lattice Z such that $Z(i, even(j)) = A_k(i, j)$ and $Z(i, odd(j)) = A_n(i, j)$.
(5) Transform I' into a stack such that elements in stack from top to bottom are values of pixels in I' in row major order.
(6) Scramble I' and search Z in column major order and if $Z(i, j) = 1$ pop an element from top of $stackI'$) into initially empty scrambled image SI at same coordinates (i, j). After search is complete search Z again in row-major order this time and if $Z(i, j) = 0$ pop an element from top of $stackI'$) into scrambled image SI at same coordinates (i, j).

With that encryption key for algorithm is CA rule used for evolving randomly generated A_0, the number of generations k used to yield A_k, and chosen value n where $(0 < n < k)$ it determines An that is combined with A_k to generate scrambling lattice Z. Generation of scrambled image SI for proposed algorithm can be expressed as encryption function e: $SI = e(I', A_0, v_{CGL}, k, n)$.

(2) Illustration of the Encryption Algorithm. Assume A_k and A_n are evolved from same initial lattice A_0. Then according to demonstrated algorithm Z is generated in the same manner as shown Figure 4 for instance.

Scrambling I' with Z gives SI as illustrated in Figure 5. Assume the values of I' pixels are different colors for now.

(3) Decryption Process. As for decryption algorithm it involves generation of scrambling lattice Z from provided keys. Steps for decryption are as follows:

(1) Generate Z from provided keys where $Z = f(A_k, A_n)$.
(2) Search Z in column major order if $Z(i, j) = 1$ then $SI(i, j)$ is added to $stack(SI)$.
(3) Search Z in row-major order if $Z(i, j) = 0$ then $SI(i, j)$ is added to $stack(SI)$.
(4) Reverse $stack(SI)$ then pop elements from stack in an initially empty lattice generating I'.

Decryption algorithm for proposed algorithm can be expressed as function d: $I' = d(SI, A_0, v_{CGL}, k, n)$.

After obtaining I' it needs to be transformed back to original grayscale version of image I. Recreation of original binary bits from Gray Code is not as straight forward as the generation process. Since pixels in grayscale can assume values ranging from 0 to 255 then any value in that range can be expressed in 8 bits maximum and this is true as well for its corresponding Gray Code version. (Algorithm 2) that demonstrates the steps required for converting Gray Code to original binary values.

3.2. Face Recognition with PCA. PCA objective is expressing points in higher dimensional space in lower dimensional subspace [45]. Satisfying this objective is done by achieving PCA goals which are according to [46] extraction of most important information from data, compression due to extraction of most important information, simplification of data description, and analysis of observations and variables structure. Steps for PCA features extraction are elaborated by [47] with Euclidean distance classifier as shown as follows:

(1) Convert 2D face images data into set of vectors as training data $\{F_1, F_2, \ldots, F_N\}$.
(2) Find average of training data by $\overline{F} = (1/2) \sum i = 1N F_i$
(3) Covariance matrix is determined with $C = (1/2) \sum i = 1N (F_i - \overline{F})(F_i - \overline{F})^T$
(4) Find eigenvectors corresponding to eigenvalues by $= \lambda V$, λ is eigenvalue and V is set of eigenvector.
(5) Image projection into eigenspace is found by $W_i = V_i^T (F_i - \overline{F})$
(6) Test image is projected with 5 and classified based on distance measured. This distance measures similarity between test image and faces database.
(7) Here Euclidian distance is used as classifier for projected data. It is found by

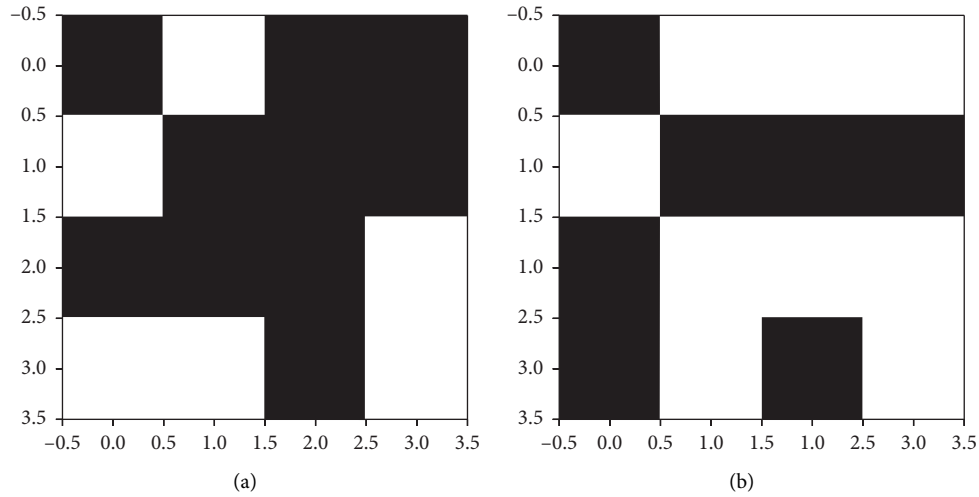

Figure 3: An example of 2D lattice (alive cells are white and dead cells are black) evolved with CGL, NM, and PBC. (a) From left initial lattice. (b) Evolved lattice.

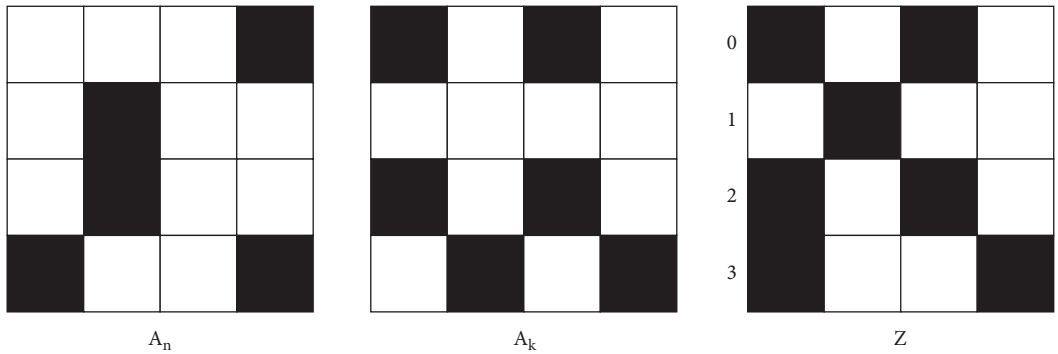

Figure 4: An example of generating Z by combining A_k and A_n.

$$d(x, y) = \sqrt[2]{\sum_{i=1}^{I} (x_i - y_i)^2}. \quad (5)$$

3.3. Encrypted Face Recognition with PCA. In order to recognize encrypted faces, PCA is used. The encrypted face images used for training and testing come from PCA, which is used to extract encrypted face images. By using encrypted images for face recognition, authentication processes can be prevented from being compromised, and credential spoofing can be prevented. Therefore, the pipeline of the face recognition scheme has been revised to include OTCA's CGL and Gray Code algorithms for image encryption. The pipeline of the proposed scheme for face recognition is shown in Figure 6. A face detection and alignment step is performed first in order to preprocess the image. The face image is then encrypted with the same key as the cipher for encrypting stored information. An encrypted image can be verified or identified by the PCA features extractor. The classification of the data and decisions are made using Euclidean distance classifiers.

4. Results and Discussion

Experiments and implementations of proposed OTCA CGL Gray Code image encryption technique and Encrypted Face PCA Recognition were conducted on Laptop with specifications 8 GB RAM, Intel(R) Core(TM) i5-3230M CPU @ 2.60 GHz and Microsoft Windows 10 Home 64 bits using Python3.

Face recognition model uses ORL faces dataset for training and testing the model. Training model uses 80% of database and 20% is used for testing. ORL dataset is already preprocessed; that is faces are already dedicated and aligned therefore preprocessing step is skipped on model implementation. Faces archive is encrypted with same key using proposed OTCA CGL Gray Code image encryption model. Using same key is better for keys management and reduces dataset encryption computational time. Using same key or different keys does not ease up or increases difficulty for distinguishing encrypted images due to high sensitivity of encryption key shown by NPCR evaluations. However, having a single key requires securing encryption key, otherwise the integrity of entire dataset could be compromised.

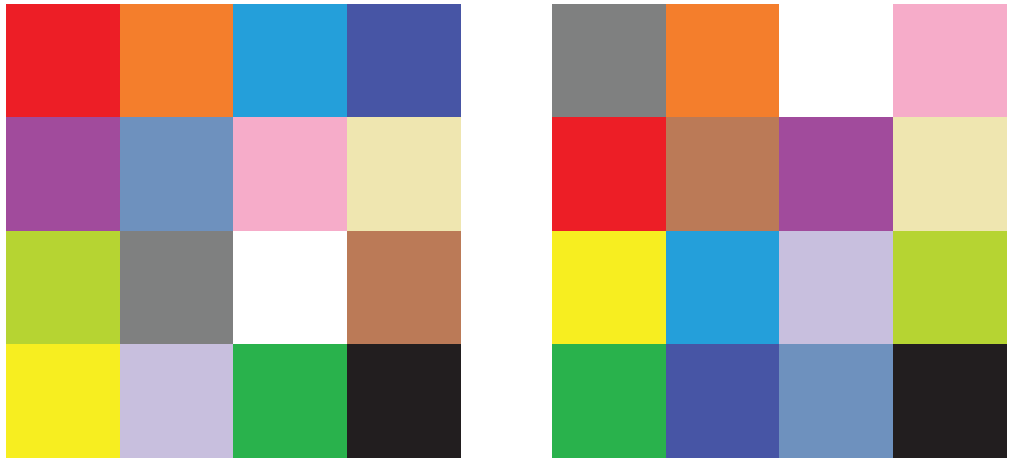

Figure 5: Generating SI by scrambling I' with Z.

4.1. Face Archive Encryption. Entire ORL faces database is encrypted with proposed image encryption method. Single encryption key is used for encryption of faces images. Figure 7 shows some of the faces encrypted with the proposed method.

Key: $SI = e(I', A_0, v_{CGL}, 13, 8)$

With that, the performance of image encryption scheme is evaluated for the histogram, correlation, number of pixels change rate (NPCR), mean absolute error (MAE), and gray difference degree (GDD). In addition a key analysis of the proposed encryption scheme is performed as well. Testing is implemented on encrypted ORL faces with the same key as in Figure 7.

4.1.1. Histogram. Histogram shows how pixels are distributed in an image and statistical characteristics as well [48–50]. Encrypted image should have a different histogram from the original image [51]. Having different histograms prevents identification of the original image from the scrambled version. The histogram represents the distribution of different pixels intensities across the image. Since Gray Code pixels substitution is applied on the image before scrambling, there will be a difference between the original image and the scrambled image histograms. This eliminates the possibility of identifying an original image from the scrambled image histogram, given that a database of original and scrambled images became available to an attacker. Figure 8 shows the comparison between original image and scrambled image histograms using a randomly selected subject image from the ORL face dataset.

4.2. Correlation. Correlation indicates similarity between original image and its scrambled version. In Table 1, correlation is calculated with Karl Pearson's formula [29]:

$$= \frac{\sum xy}{\sqrt{\sum x^2}\sqrt{\sum y^2}}, \tag{6}$$

where $x = (X - \overline{X})$ *and* $y = (Y - \overline{Y})$.

Correlation coefficient values are in the range of −1 to 1 inclusively. Having 0 correlation is a good indication for encryption robustness as it means there is no correlation between the scrambled image and the original image. Correlation values of plain images are usually closer to 1 (strong positive correlation). Value of −1 means a strong negative correlation in variables.

4.2.1. Number of Pixels Change Rate. NPCR (number of pixels change rate) finds the different percentage of pixels between the two encrypted images whose corresponding original images are different in one pixel only [24]. The higher the NPCR value, the more resilient the encryption against differential attacks [52]. The NPCR ideal value is 99.6094% [53]. NPCR is found by [54–56]. Where SI_1 is encrypted image of plain image I_1 and SI_2 is encrypted image of plain image I_2. I_2 differs from I_1 in one pixel only. Same key is used for encryption in NPCR test [57]. However, it is clear from proposed scrambling algorithm that it contains no actual diffusion stage. Diffusion stage emphasizes on establishing a dependent relationship between the encrypted image and original image in a complicated manner; where a change in one pixel in the original image changes encrypted image almost entirely [58]. As such in NPCR evaluation, a slight change is made on encryption key such that values of k, n, or both are changed. With that NPCR can be obtained as follows:

$$NPCR(SI_k, SI_{k'}) = \frac{\sum_{i=0}^{width(I)-1}\sum_{j=0}^{height(I)-1} x(i,j)}{resolution(I)},$$
$$if\ SI_k(i,j) = SI_{k'}(i,j)\ then\ x(i,j) = 0, \tag{8}$$
$$if\ SI_k(i,j) \neq SI_{k'}(i,j)\ then\ x(i,j) = 1,$$

where S_{Ik} is encrypted image with original key. $S_{Ik'}$ is encrypted image with modified k' lattice. NPCR is also

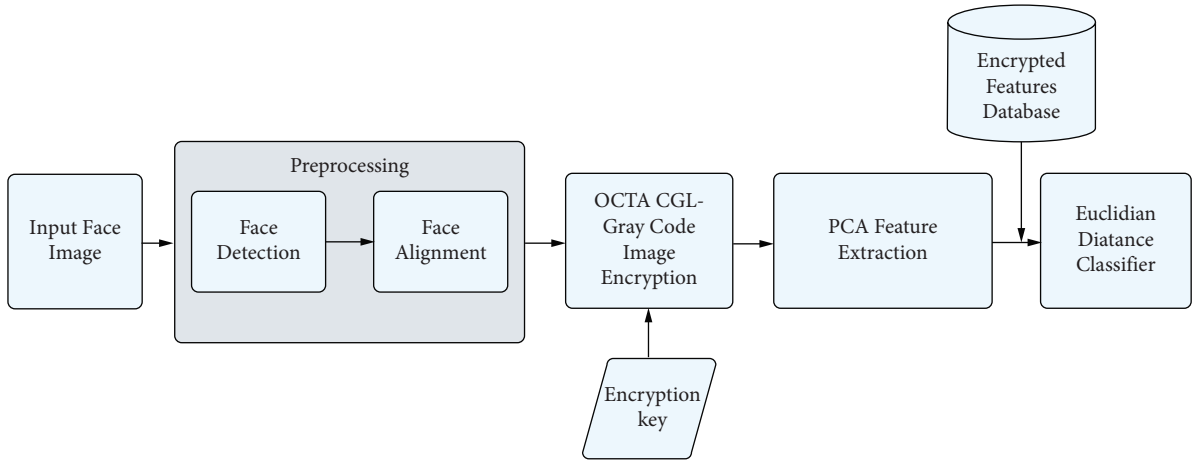

Figure 6: Encrypted face recognition pipeline.

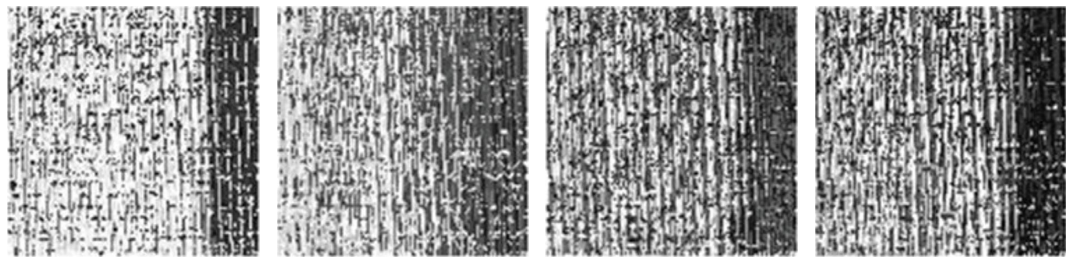

Figure 7: Sample of encrypted ORL faces taken from random subjects.

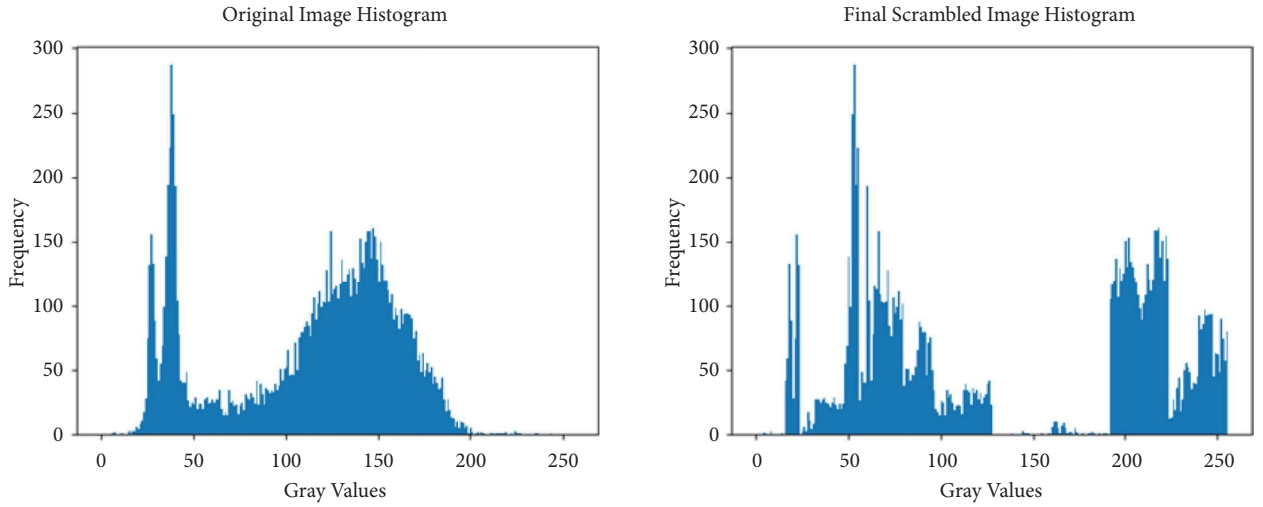

Figure 8: Histogram of randomly selected subject image before and after encryption.

utilized for finding change rate between original image and its scrambled version as in work of [29, 59]. With that NPCR between the original and scrambled image is as follows:

$$NPCR(I, SI) = \frac{\sum_{i=0}^{width(I)-1} \sum_{j=0}^{height(I)-1} x(i, j)}{resolution(I)},$$
$$if\ I(i, j) = SI(i, j)\ then\ x(i, j) = 0, \quad (9)$$
$$if\ I(i, j) \neq SI(i, j)\ then\ x(i, j) = 1.$$

NPCR is used to test percentage of pixels that change between the original image and encrypted images and to test the sensitivity of the algorithm to slight changes in keys. In the first case NPCR should be large between the original image and its encrypted version. On second case, a slight change is made on the encryption key, and two encrypted images from the same original image are tested for differences. Tables 2 and 3 show NPCR values between original and encrypted images and NPCR for sensitivity test, respectively.

Input: binary string B with length n.
Output: Gray Code representation of binary string GC.
(1) Set $n = Length(B)$
(2) Set $GC = B[0]$
(3) Set $i = 0$
(4) $while(i < n - 1)$
$GC = GC + B[i] \oplus B[i + 1]$
$i = i + 1$
(5) Return GC

Algorithm 1: Converting binary string to gray code representation.

Input: Gray Code string GC with length n.
Output: Original Binary string B
(1) Set $n = Length(GC)$
(2) Set $B = GC[0]$
(3) Set $i = 0$
(4) $while(i < n - 1)$
$B = B + B[i] \oplus GC[i + 1]$
$i = i + 1$
(5) Return B

Algorithm 2: Converting gray code to binary values.

Table 1: Correlation between randomly selected ORL images and their scrambled versions.

Test image	Correlation
$S16/8$	0.282
$S3/9$	0.142
$S12/7$	0.119
$S28/8$	0.037
$S29/2$	0.074
$S10/1$	0.158
$S35/6$	0.139
$S20/10$	0.072

Table 2: NPCR between randomly selected ORL images and their scrambled versions.

Test image	$NPCR(I, SI)$ (%)
$S15/8$	99.665
$S33/8$	99.505
$S1/5$	99.673
$S11/5$	99.537
$S23/9$	99.585
$S15/9$	99.346
$S12/10$	99.553
$S7/7$	99.649

$$NPCR(SI_1, SI_2) = \frac{\sum_{i=0}^{width(I)-1} \sum_{j=0}^{height(I)-1} x(i, j)}{resolution(I)},$$
$$if\ SI_1(i, j) = SI_2(i, j)\ then\ x(i, j) = 0,$$
$$if\ SI_1(i, j) \neq SI_2(i, j)\ then\ x(i, j) = 1. \quad (7)$$

Keys: $SI = SI_k = e(I', A_0, v_{CGL}, 13, 8)$

$$SI_{k'} = [e(I', A_0, v_{CGL}, 13, 7), e(I', A_0, v_{CGL}, 12, 8), e(I', A_0, v_{CGL}, 12, 7)]. \quad (10)$$

4.2.2. Mean Absolute Error. MAE (mean absolute error) is used to determine how different is the encrypted image from the original image [60]. It is calculated with [61] as follows:

$$MAE = \frac{1}{size(I)} \sum_{i=0}^{width(I)-1} \sum_{j=0}^{height(I)-1} |I(i, j) - SI(i, j)|. \quad (11)$$

The values of MAE are in range $[0, 2^{N-1}]$. N is number of pixels bits. Higher MAE indicates more differences

Table 3: NPCR scrambling key sensitivity test.

Test image	$NPCR(SI_k, SI_{k'})$		
	$NPCR(SI_k, SI_{k'}[0])$ (%)	$NPCR(SI_k, SI_{k'}[1])$ (%)	$NPCR(SI_k, SI_{k'}[2])$ (%)
*S*15/8	98.022	96.643	98.628
*S*33/8	98.014	96.348	98.437
*S*1/5	97.656	96.061	98.477
*S*11/5	97.720	96.372	98.254
*S*23/9	98.126	96.197	98.628
*S*15/9	97.863	96.109	98.325
*S*12/10	98.309	96.859	98.971
*S*7/7	97.823	96.301	98.557

between encrypted image and original image which is a desirable trait for encryption robustness. Table 4 shows MAE for randomly selected encrypted images.

4.2.3. Gray Difference Degree. Gray difference degree (GDD) measures performance of scrambling on an original image. Introduced by [26] GDD is calculated using the following steps:

(1) For each pixel P where $P(i, j) \in I \; and \; edge \; of \; I\,(P(i, j), I) = False$ find Gray Difference (GD) by $GD(i, j) = (1/4)\sum_{i', j'} [P(i, j) - P(i', j')]^2$ where $(i', j') \in N_{vN}(i, j, 1) if |i' - i| + |j' - j| \leq 1$,

(2) Find average neighborhood GD for all pixels in I using GDs calculated in 1 using function $Avg(GD(i, j)) = (\sum_{i=1}^{wideth(I)-2} \sum_{j=1}^{height(I)-2} GD(i, j)/(width(I) - 2) \times (height(I) - 2))$.

(3) Repeat steps 1 and 2 for SI to obtain $Avg_{SI}(GD_{SI}(i, j))$,

(4) Compute GDD using $GDD = (Avg_{SI}(GD_{SI}(i, j)) - Avg(GD(i, j))/Avg_{SI}(GD_{SI}(i, j)) + Avg(GD(i, j)))$.

Table 5 shows obtained GDD values for randomly selected subject images from ORL dataset.

Table 4: MAE values for encrypted images.

Test image	MAE
*S*9/8	68.728
*S*23/5	60.517
*S*23/6	57.734
*S*12/8	66.906
*S*29/1	43.580
*S*34/10	59.224
*S*10/3	77.005
*S*35/8	66.644

Table 5: GDD values for encrypted images.

Test image	GDD
*S*1/10	0.9753
*S*14/1	0.9350
*S*32/9	0.9491
*S*18/1	0.9582
*S*28/1	0.9469
*S*9/10	0.9682
*S*13/6	0.9502
*S*3/2	0.9687

4.2.4. Key Analysis. Encryption keys are fundamental components for implementing encryption on subject images. Encryption keys' resistance to attacks should be high. Encryption keys should have a large key space and a high sensitivity [62]. Keys with larger key space are more resistive to brute force attacks [63]. Resisting brute force attacks requires key space to be $>2^{100}$ [64]. As for keys sensitivity test, a small change in encryption key should have large difference on the generated encrypted image [65]. NPCR is utilized in key analysis test in similar manner to work of [66].

To test the effectiveness of the image encryption key space must be large enough to withstand brute-force attacks [29]. For the proposed algorithm, the key is composed of an initial A_0 lattice of size width $*$ height, number of generations k, and value n such that $0 < n < k$. Since n is selected randomly based on k, and pixels on initial A_0 lattice can assume one of two states (alive or dead) then key space is $u(2^{size(A_0)})$ where $u = (k(k-1))/2$ is the size of unique pairs of k and n set. Key space is exceptionally wide, and large enough k value (which in turn increases the size of u) can be selected to effectively encrypt images of smaller size. At minimum for an image of size (10×10) key space is $u\,(2^{100}) > 2\,100$ which exceeds the brute force resistivity limit.

To test the sensitivity of the key, a random subject encrypted image is decrypted using A_k and A_n only. Then keys with different values of k and n are tested to decrypt the image. Results on Figure 9 show that decrypting image is only possible with the correct key. Given that A_0 is available, decrypting an image with A_k or A_n only yields no useful information, and the same can be concluded for different values of k and n. Also, the NPCR test in Table 3 shows a high change rate between images encrypted with a slight change in encryption keys (>96% on minimum) which proves the high sensitivity of encryption keys.

Figure 9: Decrypting a random ORL subject image with different values of k and n. From left encrypted image, decryption with A_k only, decryption with A_n only, decryption with $k = 13$, $n = 7$, and decryption with $k = 12$, $n = 8$.

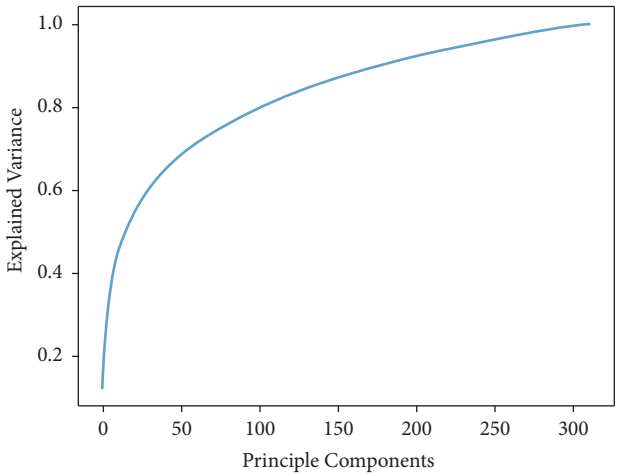

Figure 10: Explained variance by principle components.

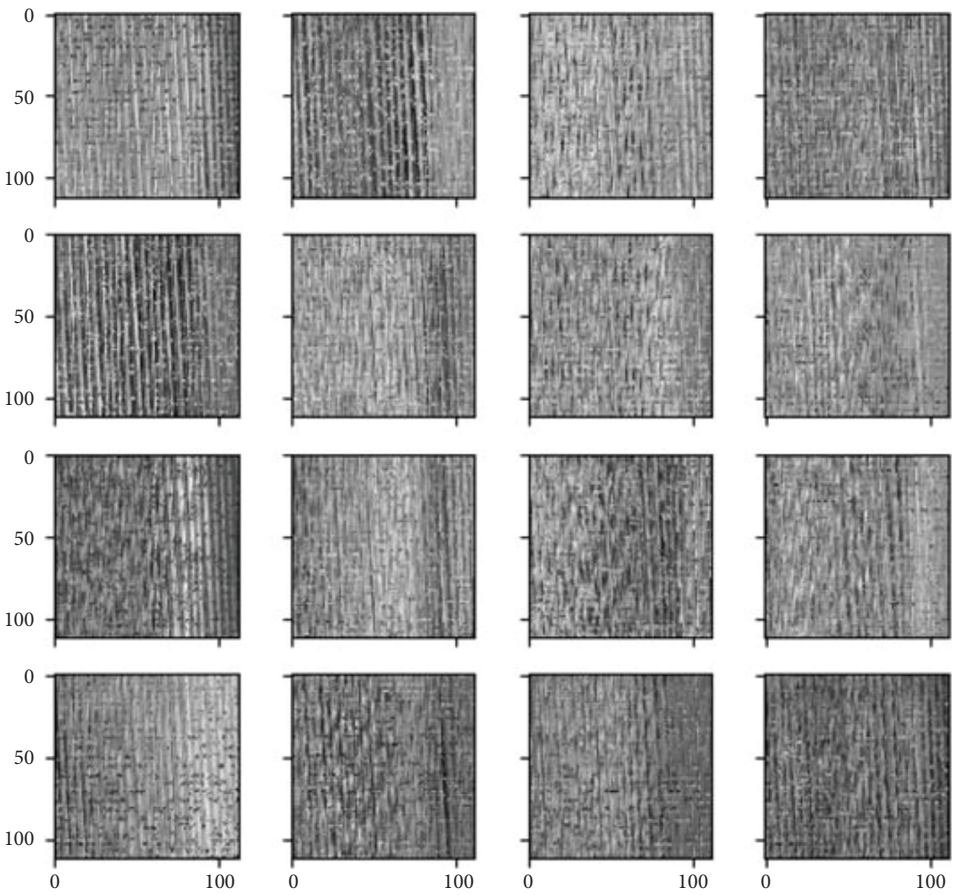

Figure 11: Eigen faces generated from encrypted face images.

Table 6: Encrypted faces classification report.

Enc. face test image	Classification result	Evaluation
S1/9	S1 with Euclidean distance 5373.228051	True
S1/10	S1 with Euclidean distance 5748.435552	True
S2/9	S2 with Euclidean distance 4473.023282	True
S2/10	S2 with Euclidean distance 3973.352329	True
S3/9	S4 with Euclidean distance 5274.349620	False
S3/10	S3 with Euclidean distance 5051.268398	True
S4/9	S4 with Euclidean distance 3942.744650	True
S4/10	S4 with Euclidean distance 4165.568529	True
S5/9	S5 with Euclidean distance 4215.311359	True
S5/10	S40 with Euclidean distance 5483.203667	False
S6/9	S6 with Euclidean distance 4393.905716	True
S6/10	S6 with Euclidean distance 2908.963330	True
S7/9	S7 with Euclidean distance 3823.492582	True
S7/10	S7 with Euclidean distance 4425.482036	True
S8/9	S8 with Euclidean distance 4497.363225	True
S8/10	S8 with Euclidean distance 3792.915907	True
S9/9	S9 with Euclidean distance 3860.873395	True
S9/10	S9 with Euclidean distance 4905.702679	True
S10/9	S8 with Euclidean distance 6003.611492	False
S10/10	S8 with Euclidean distance 5835.079534	False
S11/9	S11 with Euclidean distance 4116.744312	True
S11/10	S11 with Euclidean distance 3838.417882	True
S12/9	S12 with Euclidean distance 4765.491458	True
S12/10	S12 with Euclidean distance 5292.675978	True
S13/9	S13 with Euclidean distance 5053.542112	True
S13/10	S13 with Euclidean distance 3449.380342	True
S14/9	S14 with Euclidean distance 4563.140665	True
S14/10	S14 with Euclidean distance 5607.276628	True
S15/9	S15 with Euclidean distance 3238.105065	True
S15/10	S15 with Euclidean distance 3376.481119	True
S16/9	S16 with Euclidean distance 6413.669511	True
S16/10	S16 with Euclidean distance 4447.723711	True
S17/9	S17 with Euclidean distance 4208.686506	True
S17/10	S17 with Euclidean distance 3669.955253	True
S18/9	S18 with Euclidean distance 4666.727826	True
S18/10	S18 with Euclidean distance 5171.592818	True
S19/9	S15 with Euclidean distance 6058.791313	False
S19/10	S19 with Euclidean distance 3402.706098	True
S20/9	S20 with Euclidean distance 4136.461436	True
S20/10	S20 with Euclidean distance 3870.016443	True
S21/9	S21 with Euclidean distance 4272.902692	True
S21/10	S21 with Euclidean distance 4118.266490	True
S22/9	S22 with Euclidean distance 2636.987145	True
S22/10	S22 with Euclidean distance 3540.276387	True
S23/9	S23 with Euclidean distance 4589.453589	True
S23/10	S23 with Euclidean distance 3602.207713	True
S24/9	S24 with Euclidean distance 5087.230445	True
S24/10	S24 with Euclidean distance 4895.659062	True
S25/9	S25 with Euclidean distance 4225.148892	True
S25/10	S25 with Euclidean distance 3716.330952	True
S26/9	S26 with Euclidean distance 4325.390877	True
S26/10	S26 with Euclidean distance 3035.202458	True
S27/9	S27 with Euclidean distance 3366.186844	True
S27/10	S27 with Euclidean distance 4494.598557	True
S28/9	S28 with Euclidean distance 5812.395899	True
S28/10	S28 with Euclidean distance 4965.903614	True
S29/9	S29 with Euclidean distance 4324.022162	True
S29/10	S29 with Euclidean distance 3714.952745	True
S30/9	S30 with Euclidean distance 3970.182573	True
S30/10	S30 with Euclidean distance 4164.566874	True

Table 6: Continued.

Enc. face test image	Classification result	Evaluation
*S*31/9	*S*31 with Euclidean distance 4281.472462	True
*S*31/10	*S*31 with Euclidean distance 4148.592567	True
*S*32/9	*S*32 with Euclidean distance 5129.574047	True
*S*32/10	*S*32 with Euclidean distance 4362.000327	True
*S*33/9	*S*33 with Euclidean distance 4391.321723	True
*S*33/10	*S*33 with Euclidean distance 1855.914736	True
*S*34/9	*S*34 with Euclidean distance 4046.684219	True
*S*34/10	*S*34 with Euclidean distance 3364.459445	True
*S*35/9	*S*35 with Euclidean distance 5574.973183	True
*S*35/10	*S*35 with Euclidean distance 5012.925485	True
*S*36/9	*S*36 with Euclidean distance 4977.578999	True
*S*36/10	*S*36 with Euclidean distance 5861.877623	True
*S*37/9	*S*37 with Euclidean distance 2572.658060	True
*S*37/10	*S*37 with Euclidean distance 3477.025249	True
*S*38/9	*S*38 with Euclidean distance 3935.824694	True
*S*38/10	*S*38 with Euclidean distance 2906.621539	True
*S*39/9	*S*39 with Euclidean distance 3409.791218	True
*S*39/10	*S*39 with Euclidean distance 4106.191621	True
*S*40/9	*S*40 with Euclidean distance 5729.986808	True
*S*40/10	*S*23 with Euclidean distance 5469.463603	False

Note. Sn/y means subject number/test image name. Each subject has 2 test images named 9 and 10.

4.3. PCA Implementation. The ORL faces database contains 400 pictures of 40 different subjects. Each subject has 10 faces in various poses. Thus, face images are classified into 40 classes in the ORL database and labeled accordingly. 8 images from each subject are taken for training the PCA model, and 2 images are left for testing.

Training face images are converted to vectors, and these vectors are then processed with PCA to reduce features and generate principal components that explain most variance. The original number of features in grayscale training face images is 12544 which is to be reduced according to a selected number of principle components. Figure 10 shows the number of generated principle components versus explained variance. Initially, the first few principal components explained most variations in encrypted facial features, but as the number of principal components increased, the variance explained by subsequent principles decreased. From Figure 10, 250 components are selected as the number of principal components, which account for 96% of the variance in the extracted features. In order to distinguish encrypted faces, these 250 components are used to create eigen faces. Figure 11 shows the resulting eigen faces.

4.3.1. Classification Results. A Euclidean distance classifier is used to classify test encrypted face images into the correct class labels. Table 6 shows the classification results for testing data on extracted features.

Table 6 provides a classification report for which 74 out of 80 classifications were correct. Based on that, the model was able to identify 92.5% of the encrypted faces in the database correctly.

In the second test, decrypted images were used, and we performed the same classification test. The objective is to demonstrate that in order to perform correct facial recognition, the subject image must be encrypted with the correct key; otherwise, the result is faulty. Table 7 shows the results of the classification of decrypted face images taken from the encrypted images database.

Table 7 on the classification report shows only 3 correct identifications out of 80, an accuracy of only 0.0375%. This highlights the purpose of encrypted face recognition with PCA. A method for protecting authentication processes against spoofing attacks, so that identification requires encrypting the input image with a correct key so it can be recognized correctly.

4.3.2. Execution Time. For the identification of a single face, a script was run in python3 using a single face query. With the timeit method in Python, identification time was measured for 10,000 iterations. It took 28.0455 seconds to perform 10,000 iterations. The average execution time for a single iteration is 0.0028045 seconds or 2.8045 milliseconds (milliseconds).

Table 7: Decrypted faces classification report.

Dec. face test image	Classification result	Evaluation
S1/9	S24 with Euclidean distance 7435.847954	False
S1/10	S21 with Euclidean distance 7269.929589	False
S2/9	S29 with Euclidean distance 5365.385391	False
S2/10	S29 with Euclidean distance 5573.841664	False
S3/9	S29 with Euclidean distance 6084.440200	False
S3/10	S29 with Euclidean distance 6050.517050	False
S4/9	S29 with Euclidean distance 5450.521142	False
S4/10	S29 with Euclidean distance 6124.181955	False
S5/9	S29 with Euclidean distance 6609.509702	False
S5/10	S23 with Euclidean distance 6811.295543	False
S6/9	S15 with Euclidean distance 6793.770404	False
S6/10	S26 with Euclidean distance 6972.557627	False
S7/9	S29 with Euclidean distance 5848.651596	False
S7/10	S29 with Euclidean distance 6084.717652	False
S8/9	S22 with Euclidean distance 6256.912823	False
S8/10	S22 with Euclidean distance 6309.784462	False
S9/9	S29 with Euclidean distance 5528.818348	False
S9/10	S29 with Euclidean distance 5863.762005	False
S10/9	S29 with Euclidean distance 5343.583505	False
S10/10	S29 with Euclidean distance 6095.577903	False
S11/9	S22 with Euclidean distance 5353.236949	False
S11/10	S39 with Euclidean distance 5142.255364	False
S12/9	S29 with Euclidean distance 6191.075499	False
S12/10	S29 with Euclidean distance 6171.752094	False
S13/9	S29 with Euclidean distance 6530.273543	False
S13/10	S29 with Euclidean distance 6355.567702	False
S14/9	S39 with Euclidean distance 5167.465309	False
S14/10	S29 with Euclidean distance 5228.488687	False
S15/9	S29 with Euclidean distance 5205.926729	False
S15/10	S29 with Euclidean distance 5157.896387	False
S16/9	S29 with Euclidean distance 6243.478599	False
S16/10	S29 with Euclidean distance 5997.870417	False
S17/9	S29 with Euclidean distance 6046.541073	False
S17/10	S29 with Euclidean distance 6001.829722	False
S18/9	S23 with Euclidean distance 6924.760926	False
S18/10	S23 with Euclidean distance 6817.498247	False
S19/9	S22 with Euclidean distance 5740.461747	False
S19/10	S29 with Euclidean distance 5730.211884	False
S20/9	S29 with Euclidean distance 6113.835415	False
S20/10	S22 with Euclidean distance 4731.696099	False
S21/9	S29 with Euclidean distance 5022.505110	False
S21/10	S29 with Euclidean distance 5135.471721	False
S22/9	S22 with Euclidean distance 4561.435970	True
S22/10	S22 with Euclidean distance 4598.668918	True
S23/9	S29 with Euclidean distance 5379.050528	False
S23/10	S29 with Euclidean distance 5462.854064	False
S24/9	S22 with Euclidean distance 5599.986329	False
S24/10	S22 with Euclidean distance 5487.662594	False
S25/9	S29 with Euclidean distance 5782.152301	False
S25/10	S29 with Euclidean distance 6040.679113	False
S26/9	S29 with Euclidean distance 5831.873694	False
S26/10	S29 with Euclidean distance 6077.316626	False
S27/9	S22 with Euclidean distance 5566.636033	False
S27/10	S22 with Euclidean distance 5532.136618	False
S28/9	S29 with Euclidean distance 5560.095559	False
S28/10	S29 with Euclidean distance 5481.378496	False
S29/9	S22 with Euclidean distance 4438.010799	False
S29/10	S22 with Euclidean distance 4548.927352	False
S30/9	S22 with Euclidean distance 4994.185792	False
S30/10	S22 with Euclidean distance 4957.033765	False

Table 7: Continued.

Dec. face test image	Classification result	Evaluation
*S*31/9	*S*29 with Euclidean distance 5076.832703	False
*S*31/10	*S*29 with Euclidean distance 5404.996482	False
*S*32/9	*S*29 with Euclidean distance 5907.932649	False
*S*32/10	*S*29 with Euclidean distance 5634.050339	False
*S*33/9	*S*22 with Euclidean distance 4927.234249	False
*S*33/10	*S*22 with Euclidean distance 4480.603166	False
*S*34/9	*S*22 with Euclidean distance 4653.166250	False
*S*34/10	*S*22 with Euclidean distance 4691.198933	False
*S*35/9	*S*29 with Euclidean distance 6264.247577	False
*S*35/10	*S*29 with Euclidean distance 6216.668156	False
*S*36/9	*S*22 with Euclidean distance 5796.131605	False
*S*36/10	*S*22 with Euclidean distance 5961.330996	False
*S*37/9	*S*29 with Euclidean distance 5866.798374	False
*S*37/10	*S*29 with Euclidean distance 5856.501713	False
*S*38/9	*S*29 with Euclidean distance 5435.849055	False
*S*38/10	*S*29 with Euclidean distance 5097.791611	False
*S*39/9	*S*39 with Euclidean distance 4550.683978	True
*S*39/10	*S*22 with Euclidean distance 4515.565396	False
*S*40/9	*S*29 with Euclidean distance 6445.069443	False
*S*40/10	*S*29 with Euclidean distance 5420.880433	False

5. Conclusion

In conclusion, biometric authentication systems suffer from vulnerabilities that render such systems insecure against spoofing attacks. Those vulnerabilities extend to facial recognition systems as well. In order to resolve the spoofing issue, a new image encryption model was developed and integrated on the recognition pipeline of the PCA-based face recognition system. The image encryption model was used to encrypt the ORL face dataset used to train and test the model. With that, correct identification of face images requires encryption of the input face image with the same key used to encrypt the features database.

Testing encryption performance was carried out on randomly selected encrypted samples from the ORL dataset. Results showed that correlation was weak, histogram was different, NPCR values were >99%, MAE score was >40 on minimum and GDD values exceeded 0.92 on tested ORL face images. As for the key space, it was more than the brute force attack resistivity limit (2100), as the key space provided by the method depends on image size. Key sensitivity was tested as well, where an NPCR test was performed between face images encrypted with slightly different keys. In all cases, NPCR values were >96%.

After encrypting ORL faces dataset 80% of data was used to train the PCA model and the remaining 20% was reserved for testing. The model was able to achieve 92.5% accuracy in identifying encrypted test images from the encrypted features database. The same test was repeated again; however, this time with test face images that were entirely decrypted. In the second case, the system had an accuracy of 0.0375% in identifying decrypted images on a database of encrypted features. This shows the system's ability to withstand spoofing attacks, as the submitted input image is required to be encrypted with the correct key before it can be correctly recognized.

Data Availability

Data is available at reasonable request from the corresponding authors. Please contact Basil Ibrahim through e-mail address basilshakkak@gmail.com You may also contact Dr. Eimad Abusham eabusham@su.edu.om.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This research was supported by Sohar University . You may contact corresponding authors for any inquiries.

References

[1] L. Li, X. Mu, S. Li, and H. Peng, "A review of face recognition technology," *IEEE Access*, vol. 8, pp. 139110–139120, 2020.

[2] C. Ding and D. Tao, "Pose-invariant face recognition with homography-based normalization," *Pattern Recognition*, vol. 66, pp. 144–152, 2017.

[3] R. He, W.-S. Zheng, and B.-G. Hu, "Maximum correntropy criterion for robust face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1561–1576, 2011.

[4] F. Schroff, D. Kalenichenko, and P. James, "Facenet: a unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815–823, 2015.

[5] A. Hadid and M. Pietik¨ainen, "Combining appearance and motion for face and gender recognition from videos," *Pattern Recognition*, vol. 42, no. 11, pp. 2818–2827, 2009.

[6] R. He, W.-S. Zheng, B.-G. Hu, and X.-W. Kong, "Two-stage nonnegative sparse representation for large-scale face recognition," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 1, pp. 35–46, 2013.

[7] Y. Sun, K. Nasrollahi, Z. Sun, and T. Tan, "Complementary cohort strategy for multimodal face pair matching," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 5, pp. 937–950, 2016.

[8] H. Han, S. Shan, X. Chen, and W. Gao, "A comparative study on illumination preprocessing in face recognition," *Pattern Recognition*, vol. 46, no. 6, pp. 1691–1699, 2013.

[9] D. F. Smith, A. Wiliem, and B. C. Lovell, "Face recognition on consumer devices: reflections on replay attacks," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 736–745, 2015.

[10] X. Tan, S. Chen, Z.-H. Zhou, and F. Zhang, "Face recognition from a single image per person: a survey," *Pattern Recognition*, vol. 39, no. 9, pp. 1725–1745, 2006.

[11] R. He, Y. Cai, T. Tan, and L. Davis, "Learning predictable binary codes for face indexing," *Pattern Recognition*, vol. 48, no. 10, pp. 3160–3168, 2015.

[12] G. Guo and N. Zhang, "A survey on deep learning based face recognition," *Computer Vision and Image Understanding*, vol. 189, Article ID 102805, 2019.

[13] X. Wang, Z. Yan, R. Zhang, and P. Zhang, "Attacks and defenses in user authentication systems: a survey," *Journal of Network and Computer Applications*, vol. 188, Article ID 103080, 2021.

[14] B. Zhou, Z. Xie, and Y. Fan, "Multi-modal face authentication using deep visual and acoustic features," in *Proceedings of the ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, Shanghai, China, 20-24 May 2019.

[15] M. O. Oloyede, A. Ao, and K. S. Adewole, "Fingerprint biometric authentication for enhancing staff attendance system," *Informatics*, vol. 5, no. 3, 2013.

[16] M. Jawad Abed Al Imari, A. I. K. Al-Kaif, S. Anwar Jaafar, A. A. Hayder, and Z. Trik, "Evaluation of vitamin d level in serum blood of rheumatoid arthritis patients in babylon province," *Systematic Reviews in Pharmacy*, vol. 12, no. 1, pp. 268–271, 2021.

[17] H. Aizhar, "Dna repair genes (ape1 and xrcc1) polymorphisms–cadmium interaction in fuel station workers," *Journal of Pharmaceutical Negative Results*, vol. 13, no. 2, pp. 32–37, 2022.

[18] T. R. A. N. S. A. C. T. I. O. N. S. Tmlai, "Face spoofing and counter-spoofing: a survey of state-of-the-art algorithms," *Transactions on Machine Learning and Artificial Intelligence*, vol. 5, no. 2, 2017.

[19] P. Nagarsheth, E. Khoury, K. Patil, and M. Garland, "Replay attack detection using dnn for channel discrimination," *Interspeech*, pp. 97–101, 2017.

[20] S. Jacob, W. W. Pang, and P. S. Liang, "Certified defenses for data poisoning attacks," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[21] B. Yosefnezhad Irani, P. Ayubi, F. Amani Jabalkandi, M. Yousefi Valandar, and M. Jafari Barani, "Digital image scrambling based on a new one-dimensional coupled sine map," *Nonlinear Dynamics*, vol. 97, no. 4, pp. 2693–2721, 2019.

[22] F. Maleki, M. Ali, S. Mehdi Hashemi, and M. Ebrahim Shiri, "An image encryption system by cellular automata with memory," in *Proceedings of the 2008 Third International Conference on Availability, Reliability and Security*, pp. 1266–1271, IEEE, Barcelona, Spain, 04-07 March 2008.

[23] N. H. Packard and S. Wolfram, "Two-dimensional cellular automata," *Journal of Statistical Physics*, vol. 38, no. 5-6, pp. 901–946, 1985.

[24] M. Kaur and V. Kumar, "A comprehensive review on image encryption techniques," *Archives of Computational Methods in Engineering*, vol. 27, no. 1, pp. 15–43, 2020.

[25] A. Latif Abu Dalhoum, B. Ali Mahafzah, A. Ayyal Awwad, I. Aldhamari, A. Ortega, and M. Alfonseca, *Digital Image Scrambling Using 2d Cellular Automata*, IEEE Multimedia, 2012.

[26] R. Ye and H. Li, "A novel image scrambling and watermarking scheme based on cellular automata," in *Proceedings of the International Symposium on Electronic Commerce and Security*, pp. 938–941, IEEE, Guangzhou, China, 03-05 August 2008.

[27] Z. Jeelani and F. Qadir, "Cellular automata-based approach for digital image scrambling," *International Journal of Intelligent Computing and Cybernetics*, vol. 11, no. 3, pp. 353–370, 2018.

[28] F. Qadir, M. A. Peer, and K. A. Khan, "Digital image scrambling based on two dimensional cellular automata," *International Journal of Computer Network and Information Security*, vol. 5, no. 2, pp. 36–41, 2012.

[29] Z. Jeelani, "Digital image encryption based on chaotic cellular automata," *International Journal of Computer Vision and Image Processing*, vol. 10, no. 4, pp. 29–42, 2020.

[30] P. Ping, F. Xu, Md S. Islam Babu, X. Lv, and Y. Mao, "Image scrambling scheme based on bit-level permutation and 2-d cellular automata," in *Proceedings of the 2015 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*, pp. 413–416, IEEE, Adelaide, SA, Australia, 23-25 September 2015.

[31] A. L. Abu Dalhoum, A. Madain, and H. Hiary, "Digital image scrambling based on elementary cellular automata," *Multimedia Tools and Applications*, vol. 75, no. 24, pp. 17019–17034, 2016.

[32] Y. Kortli, M. Jridi, A. Al Falou, and M. Atri, "Face recognition systems: a survey," *Sensors*, vol. 20, no. 2, p. 342, 2020.

[33] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[34] A. Lima, H. Zen, Y. Nankaku, C. Miyajima, K. Tokuda, and T. Kitamura, "On the use of kernel pca for feature extraction in speech recognition," *IEICE - Transactions on Info and Systems*, vol. 87, no. 12, pp. 2802–2811, 2004.

[35] R. Sharma and M. S. Patterh, "A new pose invariant face recognition system using pca and anfis," *Optik*, vol. 126, no. 23, pp. 3483–3487, 2015.

[36] C. Zhou, L. Wang, Q. Zhang, and X. Wei, "Face recognition based on pca and logistic regression analysis," *Optik*, vol. 125, no. 20, pp. 5916–5919, 2014.

[37] S. Karthick, S. Selvakumarasamy, C. Arun, and P. Agrawal, "WITHDRAWN: automatic attendance monitoring system using facial recognition through feature-based methods (PCA, LDA)," *Materials Today Proceedings*, 2021.

[38] S. Wolfram, "Theory and applications of cellular automata," *World Scientific*, vol. 43, no. 12, pp. 1346–1357, 1986.

[39] D. A. Zaitsev, "A generalized neighborhood for cellular automata," *Theoretical Computer Science*, vol. 666, pp. 21–35, 2017.

[40] Z. Jeelani and F. Qadir, "A comparative study of cellular automata-based digital image scrambling techniques," *Evolving Systems*, vol. 12, no. 2, pp. 359–375, 2021.

[41] S. Torbey, "Towards a framework for intuitive programming of cellular automata," *Parallel Processing Letters*, vol. 19, no. 01, pp. 73–83, 2009.

[42] J. Conway et al., "The game of life," *Scientific American*, vol. 223, no. 4, p. 4, 1970.
[43] B. Carter, "Introduction to cellular automata and conway's game of life," in *Game of Life Cellular Automata*, pp. 1–7, Springer, 2010.
[44] B. Carter, "A note on the game of life in hexagonal and pentagonal tessellations," *COMPLEX SYSTEMS-CHAMPAIGN-*, vol. 15, no. 3, p. 245, 2005.
[45] R. Vidal, Y. Ma, and S. Sastry, "Principal component analysis," *Sasty*, pp. 25–62, 2016.
[46] H.´e Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010.
[47] E. I. Abbas, M. E. Safi, and K. S. Rijab, "Face recognition rate using different classifier methods based on pca," in *Proceedings of the 2017 International Conference on Current Research in Computer Science and Information Technology (ICCIT)*, pp. 37–40, IEEE, Sulaymaniyah, Iraq, 26-27 April 2017.
[48] S. Liu, C. Guo, and J. T. Sheridan, "A review of optical image encryption techniques," *Optics & Laser Technology*, vol. 57, pp. 327–342, 2014.
[49] I. Hussain and T. Shah, "Literature survey on nonlinear components and chaotic nonlinear components of block ciphers," *Nonlinear Dynamics*, vol. 74, no. 4, pp. 869–904, 2013.
[50] A. Anees, W. A. Khan, M. A. Gondal, and I. Hussain, "Application of mean of absolute deviation method for the selection of best nonlinear component based on video encryption," *Zeitschrift f¨ur Naturforschung A*, vol. 68, no. 6-7, pp. 479–482, 2013.
[51] X.-Y. Wang, S.-X. Gu, and Y.-Q. Zhang, "Novel image encryption algorithm based on cycle shift and chaotic system," *Optics and Lasers in Engineering*, vol. 68, pp. 126–134, 2015.
[52] A. Belazi, A. A. Abd El-Latif, and S. Belghith, "A novel image encryption scheme based on substitution-permutation network and chaos," *Signal Processing*, vol. 128, pp. 155–170, 2016.
[53] H. Fan, H. Lu, C. Zhang, M. Li, and Y. Liu, "Cryptanalysis of an image encryption algorithm based on random walk and hyperchaotic systems," *Entropy*, vol. 24, no. 1, p. 40, 2021.
[54] M. Ghebleh, A. Kanso, and D. Stevanovi´c, "A novel image encryption algorithm based on piecewise linear chaotic maps and least squares approximation," *Multimedia Tools and Applications*, vol. 77, no. 6, pp. 7305–7326, 2018.
[55] H. Liu and X. Wang, "Color image encryption using spatial bit-level permutation and high-dimension chaotic system," *Optics Communications*, vol. 284, no. 16-17, pp. 3895–3903, 2011.
[56] H. Liu, A. Kadir, and P. Gong, "A fast color image encryption scheme using one-time s-boxes based on complex chaotic system and random noise," *Optics Communications*, vol. 338, pp. 340–347, 2015.
[57] H. Khanzadi, M. Eshghi, and S. E. Borujeni, "Image encryption using random bit sequence based on chaotic maps," *Arabian Journal for Science and Engineering*, vol. 39, no. 2, pp. 1039–1047, 2014.
[58] J. Dong, G. Wu, T. Yang, and Y. Li, "The improved image scrambling algorithm for the wireless image transmission systems of uavs," *Sensors*, vol. 18, no. 10, p. 3430, 2018.
[59] K. Loukhaoukha, J. Y. Chouinard, and A. Berdai, "A secure image encryption algorithm based on rubik's cube principle," *Journal of Electrical and Computer Engineering*, vol. 2012, pp. 1–13, 2012.
[60] Y. Zhang, B. Xu, and N. Zhou, "A novel image compression–encryption hybrid algorithm based on the analysis sparse representation," *Optics Communications*, vol. 392, pp. 223–233, 2017.
[61] X. Wang and N. Guan, "Chaotic image encryption algorithm based on block theory and reversible mixed cellular automata," *Optics & Laser Technology*, vol. 132, Article ID 106501, 2020.
[62] M. Ghebleh, A. Kanso, and H. Noura, "An image encryption scheme based on irregularly decimated chaotic maps," *Signal Processing: Image Communication*, vol. 29, no. 5, pp. 618–627, 2014.
[63] X. Wang and D. Luan, "A novel image encryption algorithm using chaos and reversible cellular automata," *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 11, pp. 3075–3085, 2013.
[64] N. Khalil, A. Sarhan, and M. A. M. Alshewimy, "An efficient color/grayscale image encryption scheme based on hybrid chaotic maps," *Optics & Laser Technology*, vol. 143, Article ID 107326, 2021.
[65] X. Wang and N. Guan, "A novel chaotic image encryption algorithm based on extended zigzag confusion and rna operation," *Optics & Laser Technology*, vol. 131, Article ID 106366, 2020.
[66] T. Chen, M. Zhang, J. Wu, C. Yuen, and Y. Tong, "Image encryption and compression based on kronecker compressed sensing and elementary cellular automata scrambling," *Optics & Laser Technology*, vol. 84, pp. 118–133, 2016.