Received: 30 October 2023 | Revised: 21 May 2024 | Accepted: 3 July 2024 | Published online: 31 July 2024

RESEARCH ARTICLE

Journal of Data Science and Intelligent Systems
2024, Vol. 00(00) 1–15
DOI: 10.47852/bonviewJDSIS42021976

BON VIEW PUBLISHING

Qualitative Analysis of Cryptanalysis of Images Encrypted by Border Chaotic Cellular-Automaton-Based Model Using Segmentation Techniques

Eduardo C. Silva[1], Jaqueline N. Dorneles[1] and Danielli A. Lima[1,*]

[1]*Laboratory of Intelligent Computing, Robotics and Optimization, Science and Technology of Triângulo Mineiro Campus Patrocínio, Brazil*

Abstract: Data transactions on computers and phones have become increasingly common, with numerous entities exchanging both public and private data across expanding networks. However, for sensitive data, such as private information, transmitting it over public access networks like the Internet may pose security risks. Digital images are a particularly representative and increasingly significant data type in this context. To safeguard such data from leaks and unauthorized alterations during transmission, encryption serves as a crucial measure. Various cryptographic algorithms are available, each with distinct security requirements. Among these, encryption algorithms based on Cellular Automata offer alternative solutions characterized by their highly chaotic evolutionary capabilities, providing a significant degree of obfuscation, a critical aspect for maintaining confidentiality. This study employs image segmentation techniques such as *k*-means clustering, edge detection, and thresholding as tools for cryptanalysis. These segmentation techniques are applied to digital images generated using the Border Chaotic Cellular Automata (BCCA) cryptographic model. This approach enables the identification of artifacts and the highlighting of original image features within low-quality encryption. Conversely, more robustly encrypted images, achieved through additional encryption steps, remain unaffected by these techniques, underscoring the level of confidentiality inherent in the BCCA model.

Keywords: encryption, cryptographic algorithms, cellular automata, cryptanalysis, confidentiality, digital image processing, data transactions

1. Introduction

Nowadays, the growing popularity of web-based environments and increased accessibility to tools that work using communication features and file sharing on computer networks enhances in the same proportion. A constantly used and increasingly significant file format is digital image [1]. In addition to faithfully reproducing a copy of the object described, digital images may have metadata that complements additional information. Although there are environments where digital images are constantly shared, for example, social networks, it is possible to observe that even in these environments there are various levels of restriction, for example, satellite images [2, 3], telecommunications [4, 5], health domains [6, 7], or Internet of Things (IoT) approaches [8, 9]. In this way, confidentiality is a required information security requirement, especially when dealing with sensitive data or images that need to be protected from unauthorized access and disclosure.

Cryptography encompasses a vast array of techniques designed to ensure the secure communication of information between entities in network channels [10]. In general, cryptography models are devised to systematically obfuscate information through intricate combinations of algorithms and security keys. In this case, these components, known as the encryption procedure and the accompanying data, respectively, facilitate the application of encryption and the subsequent decryption process. The files that undergo encryption during this procedure are commonly referred to as ciphertext. It is worth noting that the history of cryptography dates back to antiquity, with early examples like the Caesar cipher illustrating its ancient origins [11].

In the modern era, cryptographic models have become intrinsically linked to both software and hardware solutions that meticulously implement their encryption procedures [12]. Consequently, in order to offer enhanced security and simultaneously reduce execution costs, a multitude of innovative models have emerged, vying to outperform their predecessors in terms of speed and the capability to generate robustly encrypted

*__Corresponding author:__ Danielli A. Lima, Laboratory of Intelligent Computing, Robotics and Optimization, Science and Technology of Triângulo Mineiro Campus Patrocínio, Brazil. Email: danielli@iftm.edu.br

images or text. These novel propositions often draw inspiration from alternative models of computation, as they seek to push the boundaries of cryptographic innovation [13].

Cellular Automata (CA) are computational structures of finite size represented by cells that can undergo evolution based on transition rules. These rules govern state changes between CA cells, and this dynamic behavior can exhibit intriguing chaos, making CAs applicable in various domains. Examples of CA applications encompass the simulation of natural patterns, fire propagation models, and pedestrian evacuation simulations [14].

In the realm of cryptography, CA-based models have been developed, including those introduced by reference [15] and the Border Chaotic Cellular Automata (BCCA) model by Silva [12]. These models leverage CA's chaotic properties for secure data transformation. CA offer the benefits of simple computational representation and parallel data processing. Independent computation of CA cells enables efficient utilization of multiple processors, boosting computational speed, and scalability in cryptography. Furthermore, CA's inherent capacity for producing complex and unpredictable sequences is invaluable in cryptographic contexts. This makes CA a promising avenue for creating secure encryption and decryption methods that safeguard sensitive data's confidentiality and integrity.

The objective of this study was twofold. Firstly, it aimed to demonstrate the robustness analysis of the encrypted digital image generated in the BCCA model, proposed in previous work [12], using image segmentation techniques, including edge detection, thresholding, and the *k*-means algorithm. This approach paralleled the methodology employed in the study by reference [16]. Additionally, the study sought to evaluate the BCCA model and the images encrypted by it through qualitative analysis using image segmentation techniques. It is necessary to emphasize that our contribution lies in the comprehensive analysis and evaluation of the existing BCCA encryption model using various image segmentation techniques. Our study aims to expand its applicability and effectiveness by combining it with different image segmentation approaches.

Traditionally, articles in the field of cryptography often rely on the National Institute of Standards and Technology (NIST) methodology for evaluation [17–21]. However, when dealing with image encryption, it is crucial to consider the unique characteristics of visual data. Unlike plain text, images are inherently two-dimensional and require specialized techniques for analysis and validation. By integrating NIST standards with image segmentation methods, we aim to provide a more comprehensive evaluation of the performance of the BCCA encryption model. Similar to the approach taken by reference [22], who integrated NIST with the Gordon–Loeb Model to analyze cipher, our study aims to combine NIST standards with image segmentation methods to enhance the evaluation of the BCCA encryption model's performance.

We believe this approach provides valuable insights into the strengths and limitations of the BCCA method in the context of image encryption. Our goal is to contribute to the advancement of cryptographic techniques tailored specifically for visual data, ultimately improving the security and reliability of image encryption algorithms. This evaluation aimed to identify and assess various aspects, including borders, masks, noises, and any remnants of the original image that might persist in the encrypted version. It was essential to ensure that a sufficient number of steps were employed in the encryption process to maintain the integrity and confidentiality of the images.

2. Related Works

This section serves to provide essential definitions and key references that are integral to comprehending the proposed work. It will commence with an introduction to CA and their various applications. Following this, we will delve into notable works that employ CA as a foundation for cryptographic models. Furthermore, we will delve into image processing techniques, specifically segmentation and filtering, highlighting their relevance to the qualitative cryptanalysis results. These techniques play a crucial role in assessing the quality and security of images encrypted using the BCCA algorithm [12], which stands as a central focus of our study.

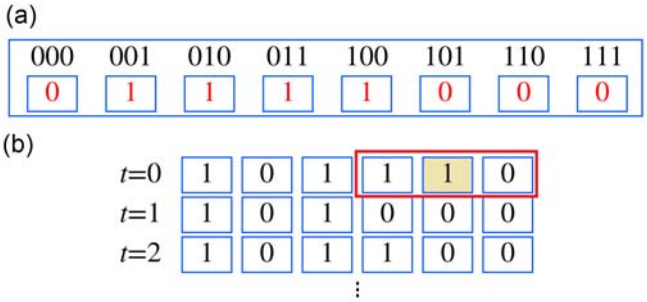
2.1. Cellular automata

A CA is composed of a cross-linked *d*-dimension divided into cells or processing units, where each cell x_i is represented by a state. Cells modify their states at each iteration step according to a transition rule. We can apply the transition rule by T steps to obtain CA lattice space-time evolution. The rule established by a transition function indicates the new symbol to be written into the lattice cell according to its current state and the states of its η^m neighbors (local rule). In its most usual definition, the states updating occurs synchronously and uses a deterministic rule, namely, at each step all the L lattice cells are updated. In the model proposed in this work, the rules system is probabilistic, that is, it is possible to change the state of a cell from a probability function, this means that the update rules are probabilistic, and therefore, the state of a cell at a given step depends probabilistically on the states of its neighboring cells in the immediately preceding time [12].

The unidimensional CA structure is the most studied. For a CA with deterministic update rule, the change of a cell state depends on m neighboring state expressed by $m = (2r + 1)$, where r is the CA radius [12]. To illustrate a one-dimensional CA with deterministic update rule, suppose a CA that addresses a model known as rule 30 [23], and also containing a 6-cell lattice and the initial state of each cell is presented in $t = 0$ as shown in Figure 1. A binary rule of radius 1 is applied, and the neighborhood of each cell is formed by three elements: the cell itself and its two adjacent neighbors (left and right). Since this CA is binary (2 possible states), there are 8 different neighborhoods, from 000 to 111 (000 0, 001 1, 010 1, 011 1, 100 1, 101 0, 110 0, 111 0). The rule itself is given by the 8 output bits associated with each possible neighborhood: 01111000.

Two-dimensional CA are widely known in the literature. Two-dimensional CA are also widely used to represent graphic patterns,

Figure 1
(a) Transition rule with radius 1. (b) The CA forward evolution $T = 2$ steps



since their execution resembles a pixelated image. This dimension contemplates several forms cell neighborhood considerations, and the two main are von Neumann and Moore neighborhood. Among the applications of two-dimensional CA, the best-known model is the mathematical game proposed by Conway in 1970 called "Game of Life."

The CA are considered versatile, finding applications in fractal, oil and fire spread modeling, pedestrian evacuation, physical systems, robotics, and cryptography. Cryptographic CA models demand specific properties. A key advantage is their computational simplicity, eliminating the need for intricate differential equations and numerical simulations. Moreover, CA can harness high parallelism, leveraging hardware such as Field Programmable Gate Arrays and Graphics Processing Units.

Cryptography leverages CA properties, including the crucial feature of sensitivity. A rule is considered sensitive when modifying an outer neighborhood bit invariably results in an output bit change. Specifically, if a change at the far-left neighborhood bit influences output bits, it exhibits left sensitivity. Otherwise we have right sensitivity. The rule applied in Figure 1(a) shows sensitivity to the left; for example, the neighborhood 000 takes the state of the central cell to 0, while the neighborhood 100 leads to 1. The rule in Figure 1 displays left sensitivity as the output complements all four pairs of analogous neighborhoods, distinguished solely by the first rule bit. This cryptographic method is part of a family that employs sensitive CA rules and computes pre-images during the encryption phase [13, 15, 24, 25].

2.2. Encryption models based on CA

This section will present some definitions and works of fundamental importance for the understanding herein work. Initially, the concept of CA and their applications will be presented. Subsequently, some works using CA modeling for the encryption will be presented and contextualized until BCCA model.

The literature has several cryptographic algorithms, such as Data Encryption Standard and Advanced Encryption Standard (AES). Both work with the proposal to make confidential information (plain text) ineligible (ciphertext). The key is used as a parameter in the encryption and decryption steps. As an alternative to classical algorithms, CA can be configured in cryptographic models with high parallel processing capacity. This feature becomes CA more relevant with the rise of processors with parallel units. There are several studies on CA and cryptography which will be seen, detailed, and discussed below.

The first known work involving the use of CA to perform cryptography was proposed by reference [23]. In this precursor model, the transition rule is fixed and presents chaotic dynamics [23]. The key is used as the initial grid from which the rule is applied by a fixed number of steps. The models that use CA in cryptography with sensitive rules have been exploited in references [26–29]; however, the parallelism and security of these models are limited due to the additive property of the rules, which do not make them chaotic. Good-quality cryptography systems should return scrambled and chaotic images [23, 30], making cryptanalysis difficult.

For a better understanding of the algorithms based on the CA pre-image calculus, some definitions will be introduced. The plain text presented in this work refers to any sequence of bits given as input, a text, or image with bits. This plain text refers to the initial L CA grid. The encrypted text is the final bit sequence found (edge, extra bits, and the itself final lattice $L^{'}$), after calculating CA pre-image through key s for T time steps. The cryptographic key set refers to the CA update rules. In this case, it is the set of all rules with sensitivity to one of the neighborhood extremes and chaotic (high entropy) that can be used to achieve robust cryptographic encryption, ensuring that even minor changes in the input lead to significant and unpredictable changes in the output. The decryption process refers to the forward evolution of the CA over with the key s for T time steps.

Encryption using CA pre-imaging emerged when researchers noticed that applying T time steps to lattice cells led to chaotic configuration sequences. CA evolution can be executed via the transition function application (forward evolution) or pre-image computation (backward evolution). In forward evolution, the lattice's initial configuration at time t evolves one time step, resulting in a new lattice configuration at time $t+1$. This procedure can be repeated for as many time steps as necessary. The backward evolution is obtained from an initial configuration of the lattice L at the instant t, the purpose is to find which lattice $L^{'}$ at time t 1 can give rise to the lattice L in the instant t, after applying the transition rule. In this case, this process can be reiterated for T time steps. A CA is deemed reversible if it possesses a unique pre-image for all feasible lattices. CA pre-image computation offers a notable benefit, which is parallelizability when there is a parallel processing unit for computing each cell, as mentioned by reference [31].

An early model implementing pre-image calculation with sensitive rules and chaotic dynamics was introduced by reference [15]. In this model, the transition rule serves as the cryptographic key, the initial lattice is defined by the original text, and consecutive pre-image calculations correspond to the encryption process. Decryption is achieved through the standard CA evolution (forward evolution). To ensure pre-image existence for any lattice, rules with sensitivity properties are utilized. Nevertheless, this approach produces ciphertext larger than the plaintext. The encryption method proposed by reference [15] will be detailed and the same is illustrated in Figure 2. To demonstrate this method, rule 30 (cipher key) with sensitivity on the left will be used. At each step, it will be necessary to start the m 1 bits to the right of the new lattice (represented in Figure 2 as blue stained cells). Once the bits have been initialized, the bitwise transition stage begins along with the neighborhood value generated by synchronously and parallelly filling all the other left lattice cells. In addition, we can start to fill in time instants $t + i$, with $i > 0$, since the dependent bits of $t + i$ have already been calculated. This procedure greatly accelerates the process, since each cell is calculated independently of the others. The process is based on the following process: given a neighborhood ?01 1 (? is the red bit, 01 are the bits in blue, and 1 is the green bit) in step $t = 1$, it attaches itself in the cell adjacent to the already calculated chain, the extreme left neighborhood bit that causes output 1 in $t = 0$. Therefore, the value found in the example will be 0, since the rule 001 1 (that is, 001 is the neighborhood that generates the exit value 1). The pre-image calculation will run until process reaches the steps T stipulated. A model disadvantage is the disturb propagation, which in this case happened only on the sensitivity rule side. It is known that a good cryptographic method must propagate the perturbation along the entire lattice.

In reference [32], a model is introduced that enlarges the lattice size. Unlike the reference [15] model, this approach utilizes rules with bidirectional sensitivity, effectively spreading bit perturbations across the entire lattice. In reference [24], researchers explored a method that maintains the lattice size, ensuring the ciphertext matches the original size. However, it comes with the drawback of being unable to encrypt all provided input text. In reference [25], an approach was developed to expand plaintext only when necessary, but it disrupts parallelism by employing global and local storage stacks for potential pre-images.

Figure 2
Methodology applied to the left sensitive rule encryption in the [15] model

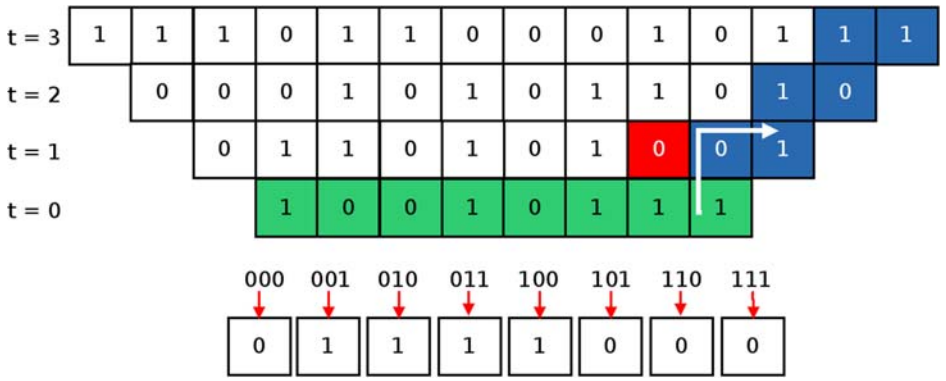


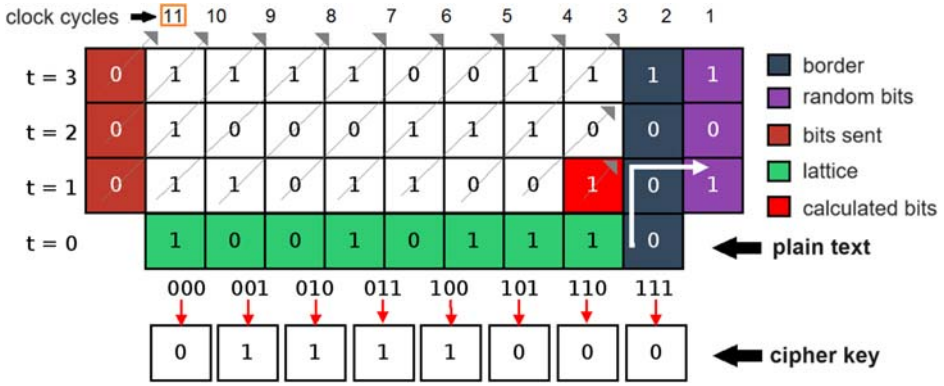
Table 1
Strengths and weaknesses of each study

Study	Strengths	Weaknesses
Gutowitz [15]	– Introduced an early model implementing pre-image calculation with sensitive rules and chaotic dynamics. – Utilized rules with sensitivity properties to ensure pre-image existence for any lattice. – Demonstrated the encryption process using rule 30 with sensitivity on the left, providing a detailed methodology. – Accelerated the encryption process by calculating each cell independently.	– Produced ciphertext larger than the plaintext, indicating inefficiency in terms of space utilization. – Disturb propagation was limited to the sensitivity rule side, rather than spreading across the entire lattice as desired.
Oliveira et al. [32]	– Introduced a model that enlarges the lattice size and spreads bit perturbations across the entire lattice using rules with bidirectional sensitivity.	– A limited rule set satisfies bidirectional CA rules.
Lima [24]	– Explored a method that maintains the lattice size, ensuring the ciphertext matches the original size.	– Unable to encrypt all provided input text, indicating limitations in applicability.
Oliveira et al. [25]	– Developed an approach to expand plaintext only when necessary, minimizing disruption to parallelism.	– Disrupted parallelism by employing global and local storage stacks for potential pre-images, potentially impacting efficiency and scalability.
Silva et al. [12]	– Introduced the BCCA method, which requires fewer. – Ensured all text inputs yield valid pre-images, enhancing reliability and applicability. – Demonstrated the high parallelizability of the BCCA method, contrasting with stack-dependent memory control models. – Security demonstrated in the Hybrid Cellular Automata model, with various evaluations and analyses conducted.	– Conducted experiments using perturbation, entropy, and histogram distribution.
Lira et al. [13]	– System detailed herein uses a hybrid mechanism to attain reversibility, and this approach is adapted to create a novel block cipher. – Fit very well in the current computational paradigm where multithreading potential is very desirable.	– Cryptographic robustness was empirically evaluated through avalanche property compliance and the NIST randomness suite, but not evaluated in other tests.

Various models also incorporated 2-dimensional and 3-dimensional CA techniques. In the model under investigation, extra bits are required but in smaller quantities compared to references [15, 32]. All text inputs yield valid pre-images, unlike reference [24]. Furthermore, the BCCA method from reference [12] is highly parallelizable, contrasting with the stack-dependent memory control model in reference [25]. Its security is demonstrated in the Hybrid CA model in reference [13], where authors conducted various evaluations and analyses, including theoretical discussions on reversibility and a graph theory-based analysis.

Table 1 shows a comparative analysis of the strengths and weaknesses of each study related to CA-based encryption models.

Figure 3
Exemplification of the encryption process of the method in reference [12]



Reference [15] introduced an early model incorporating pre-image calculation with sensitive rules and chaotic dynamics, yet it exhibited inefficiencies in terms of space utilization and limited disturbance propagation. Reference [32] introduced a model that enlarged the lattice size and spread bit perturbations across the entire lattice, although there were limitations in the available rule set. Reference [24] explored a method maintaining lattice size but faced limitations in encrypting all provided input text. Reference [25] developed an approach to expand plaintext only when necessary but disrupted parallelism with global and local storage stacks. Reference [12] introduced the BCCA method with fewer extra bits and ensured all text inputs yielded valid pre-images, demonstrating high parallelizability and security. [?] presented a hybrid mechanism for reversibility but lacked comprehensive empirical evaluation beyond certain tests. Overall, the table provides valuable insights into the strengths and weaknesses of each CA-based encryption model, facilitating a deeper understanding of their applicability and effectiveness.

2.3. BCCA encryption model

The BCCA encryption model is a method for the encryption of one-dimensional bit blocks proposed in reference [12]. A black and white image is an array of bits, where each line is a sequence of bits. The BCCA consists of the backward evolution of the CA and it is obtained from a lattice L initial configuration, which is considered the plain text, in the t step. The purpose is to find out which lattice L' in the instant t 1 can give rise to the lattice L in the instant t, after applying the transition rule, which represents the cryptographic key. This procedure can be repeated for T steps resulting in a chaotic pattern (ciphertext). To avoid that the ciphertext size increases like in reference [15], in reference [12] created a fixed border, which is responsible for limiting these bits growth. Any pseudo-random sequence can be used in the composition of that edge, including the sequence proposed by reference [23]. However, in BCCA, the edge refers to the bit sequence that corresponds to half the cryptographic key s of $r = 1$ (or any other radius). Figure 3 presents the cryptographic key given by 01111000 and the dark blue border $b = 1000$, which is half of this cryptographic key. If more bits need to be sent, they will be concatenated to the edge bit sequence b. The next bit to be concatenated is the bit 0, and the edge becomes 01000. The border must be stored and sent next to the ciphertext.

The process of encrypting a line $L = 10010111$ of the black and white image with $N = 8$ pixels is shown in Figure 3. To detail the procedure, an example will be presented for updating the bit in red at time $t = 1$, considering the completion of bit. According to the following neighborhood ?01->0. From the transition rule used as a cryptographic key, we have 101 0. Thus, the first bit is updated and all other, starting from the same definition. To apply this method to images, they must be broken into one-dimensional blocks. The blocks are linear and represent a row or an entire column of bits (pixels) of the image.

In this way, the method can be applied to each image block for T steps. The processing cost P_{CB} for performing the image block scrambling of size N is carried out in $P_{CB} = T + N + 2$ $(r$ 1) clocks cycles, where $N = \max(m, n)$ (maximum size between image columns and rows), such that $N = n$. In the example of Figure 3, we have $P_{CB} = 3 + N + 2$ (1 1) = 11 clock cycles if there are $T = 3$ processing cores units). Encryption of n m image is performed at a processing cost of $P_{CI} = m$ $(T + N + 2(r$ 1) computer clocks cycles.

If a AES image scrambling, with the same dimensions, is compared with the BCCA model, the time measured would be $P_{CB} = T$ $(2\ N + N^2 + N^3))$ clock cycles. For the computation time of an image, we would have $P_{CI} = m$ $(T$ $(2\ N + N^2 + N^3))$, such that T is the number of rounds required to encrypt a block of N bits size [33] – in this case, a line of the image. Thus, for the row's cipher in Figure 3 we have that $P_{CB} = 3$ $(2\ 8 + 8^2 + 8^3) = 3{,}288$ clocks. This delay is due to the fact that all blocks are concatenated through a XOR () function.

The decryption method, adopted in this work, is based on the evolution of a CA and is performed with the direct application of the transition function in the forward evolution of the CA next to the ciphertext and the border (stored), for T time steps. In this way, the plain text is obtained again. The processing cost P_{DB} for performing the deciphering of a block of size N of the image is carried out in $P_{DB} = T$ cycles of clock, where $N = \max(m, n)$ and let us assume that $N = n$. In the example of Figure 3, we have that the line of size N would be deciphered in $P_{DB} = 3$, if there are 8 processing cores – one for each cell of the network N. The complete deciphering of a size image $n \times m$ is performed at a processing cost of $P_{DI} = m \times T$ clock cycles of the computer.

2.4. Image processing techniques

To assess the robustness quality of digital images encrypted using the BCCA method, three image segmentation techniques were employed, utilizing the original image in Figure 4(a), with

Figure 4
Exemplification of image being processed through different segmentation techniques

Figure 5
Exemplification of image being processed through segmentation technique *k*-means using different sizes of clusters

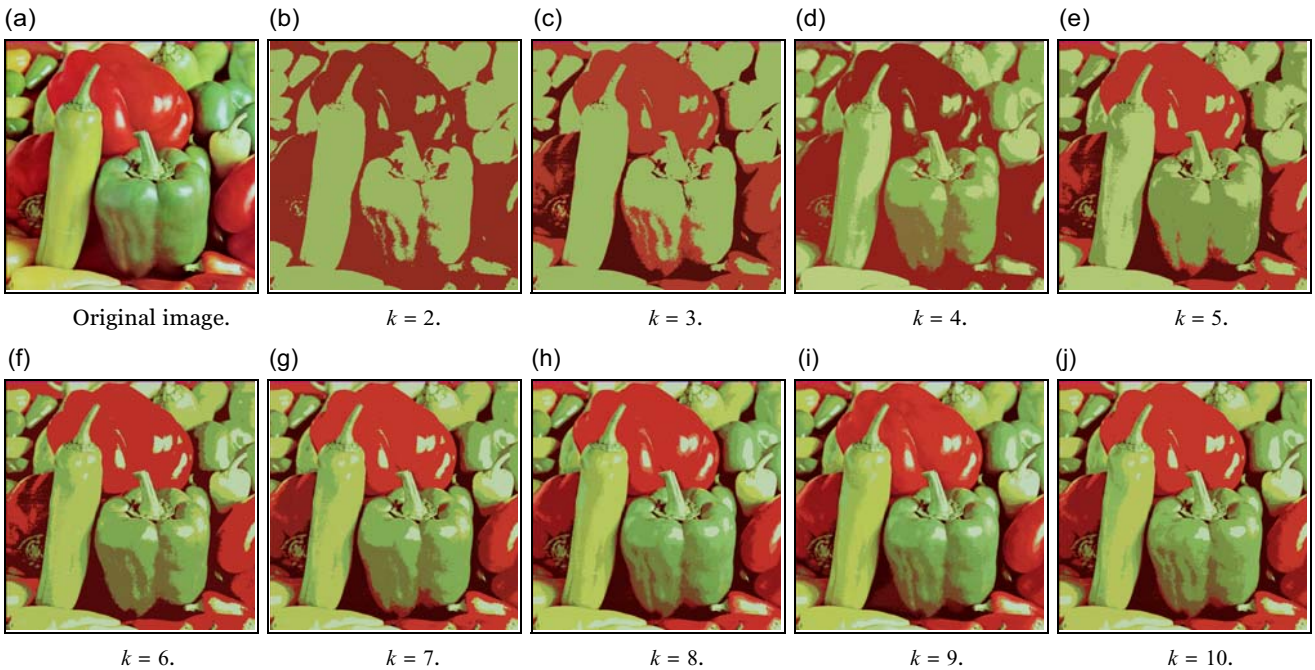
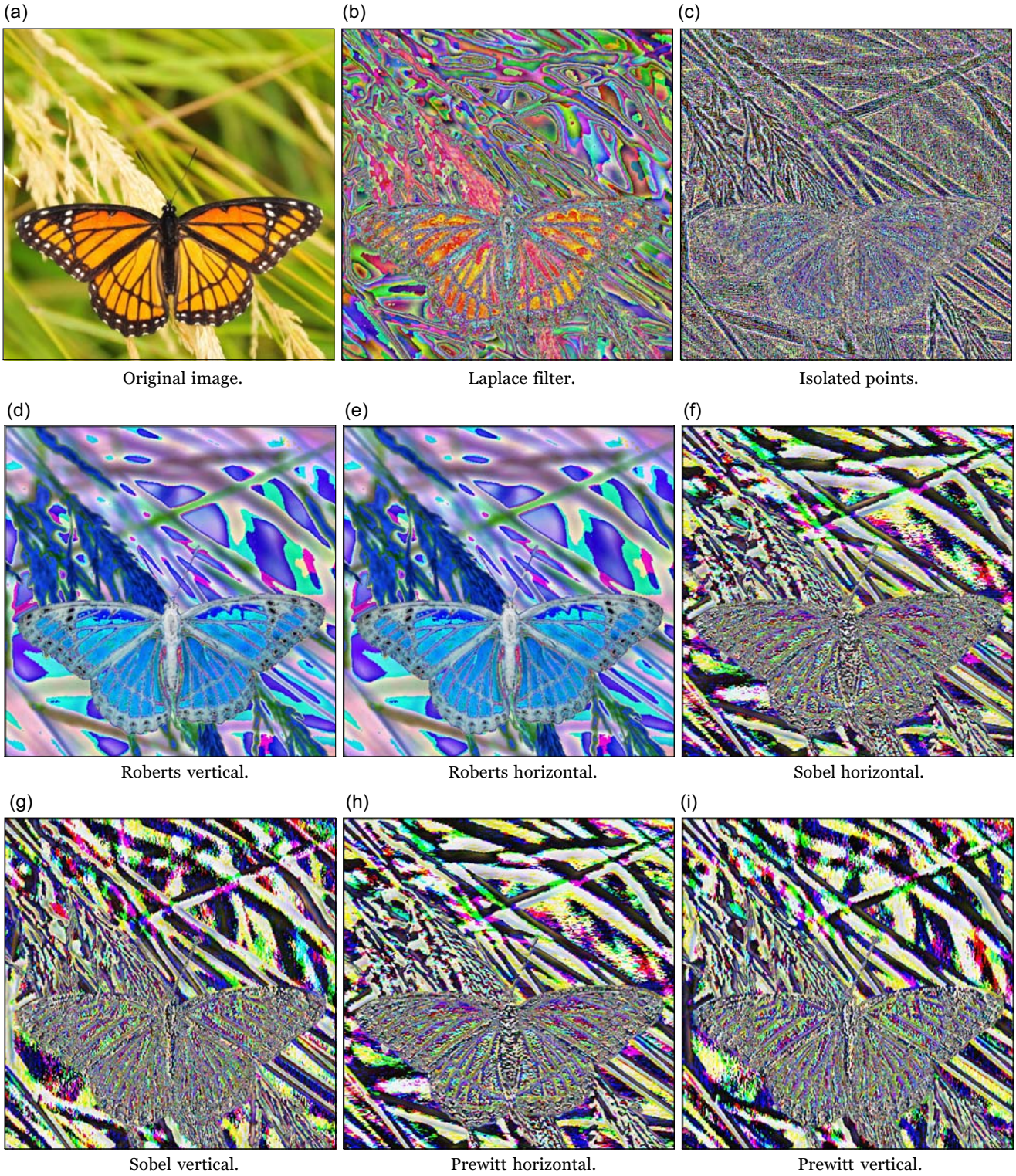


Figure 6
Exemplification of butterfly image being processed through the segmentation technique called edge detection using different masks

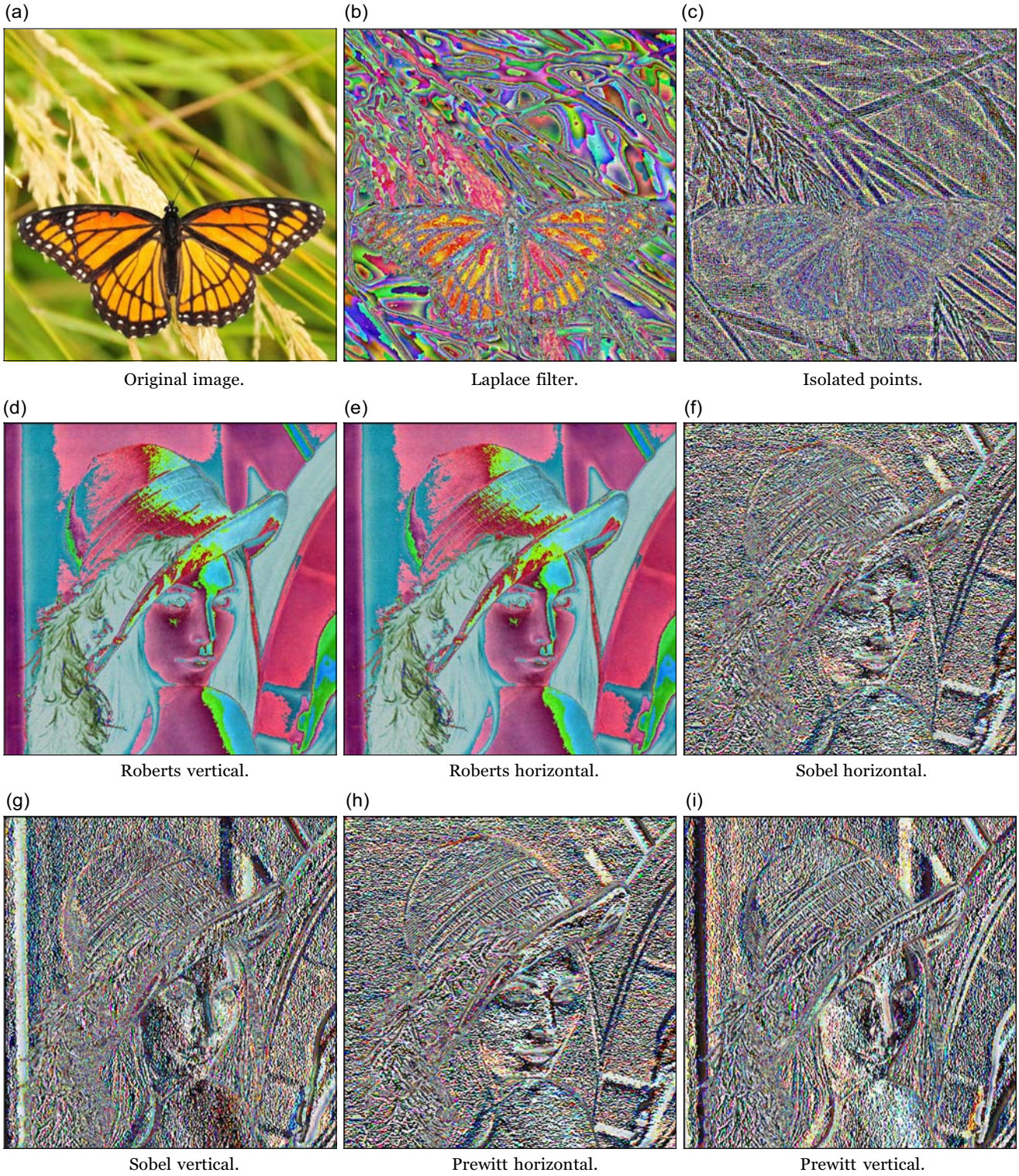


512 512 pixels. The segmentation methods employed included thresholding (i), edge detection (ii), and clustering (iii), and each is elucidated below:

Thresholding technique is generally used to separate objects from the background and hence highlight relevant areas – illustrated in Figure 4(b). When defining a threshold λ, each pixel is compared with the value of λ to define whether the future pixel descriptor will be black or white [34].

Edge detection is applied over the pixels of an image to identify variations and unify patterns, as shown in Figure 6(c). The technique

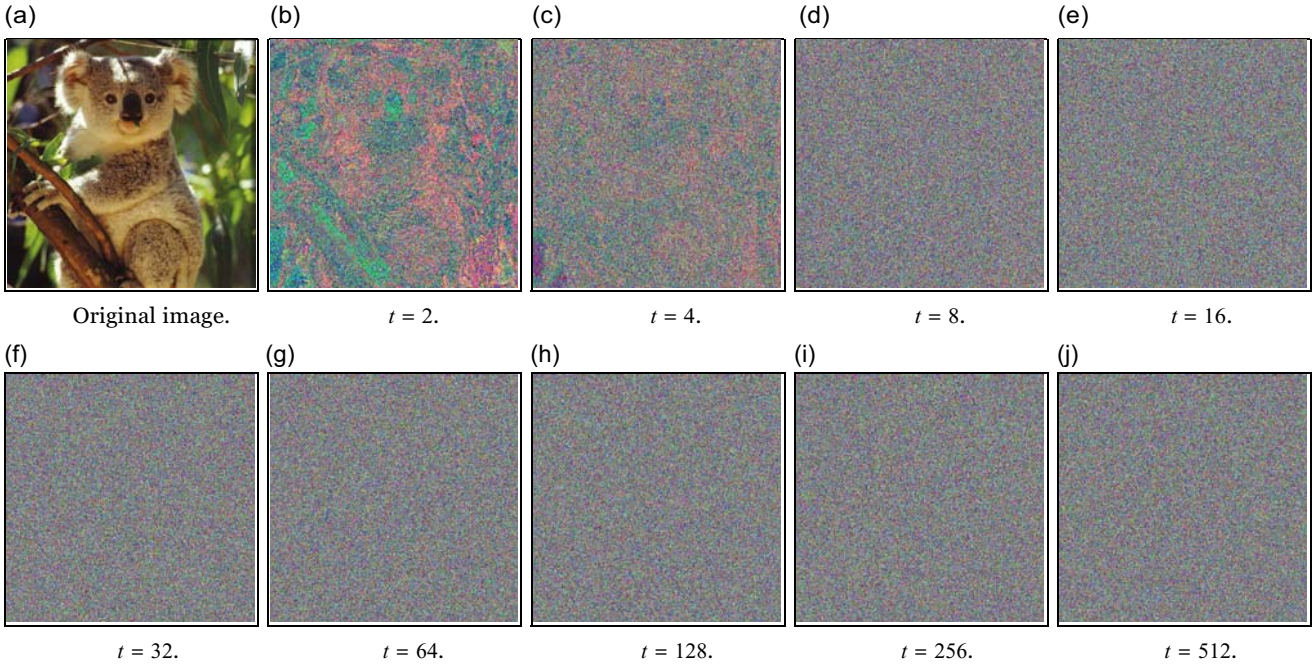
Figure 7
Exemplification of woman image being processed through the segmentation technique called edge detection using different masks



is able to highlight contours by using edge-detecting masks – boundaries between regions with distinct gray levels.

Low pass filter allows lower-frequency components of an image to pass through while reducing or attenuating the higher-frequency components, and it is often used to smooth or blur an image. It achieves this by averaging the pixel values in the neighborhood of each pixel in the image. The result is a reduction in high-frequency details and noise, resulting in a more homogeneous and less detailed appearance, as shown in Figure 6(d) ($i = 1$) to Figure 6(f) ($i = 5$). It can also be useful when you want to emphasize the broader structural elements of an image while de-emphasizing fine details.

Figure 8
Image encryption using the BCCA method from reference [12] with varying pre-image steps



Clustering using the k-means algorithm is used to group pixels that present close values; consequently, it generates an image with k subsets. This approach facilitates the creation of a uniform division that accentuates both the distinctions and resemblances among the objects within the image, as exemplified in Figure 5(a) (original image), 5(b) ($k = 2$), and extends to 5(j) ($k = 10$).

In Figure 6, an exemplification of a butterfly image being processed through the segmentation technique known as edge detection using various masks is presented. The figure comprises multiple subfigures, each demonstrating a distinct result of the edge detection process applied to the original image. Subfigure 6(a) displays the original, unprocessed image, while the subsequent subfigures 6(b) through 6(h) showcase the outcomes of edge detection utilizing different masks. These masks, including Laplace, isolated points, Roberts vertical and horizontal, Sobel horizontal and vertical, and Prewitt horizontal and vertical, reveal the impact of various edge detection techniques on the image, enhancing our understanding of their applications and capabilities.

Laplace filter is used to identify regions in an image where the intensity changes abruptly, highlighting edges and fine details. It computes the second derivative of the image to emphasize sudden transitions.

Isolated points focus on detecting isolated or individual points of high contrast within an image. It is particularly useful for spotting small, distinct features or noise.

Roberts vertical and horizontal filters are simple, basic, and quick edge detection filters. The vertical and horizontal versions emphasize changes in intensity in these respective directions.

Sobel horizontal and vertical filters are more complex and emphasize edges more effectively than the Roberts filters. They are widely used for edge detection, as they take into account the gradient in both horizontal and vertical directions.

Prewitt horizontal and vertical filters are similar to the Sobel filters, and the Prewitt filters are used for edge detection. They calculate gradients in both the horizontal and vertical directions, aiding in the identification of edges and contours in an image.

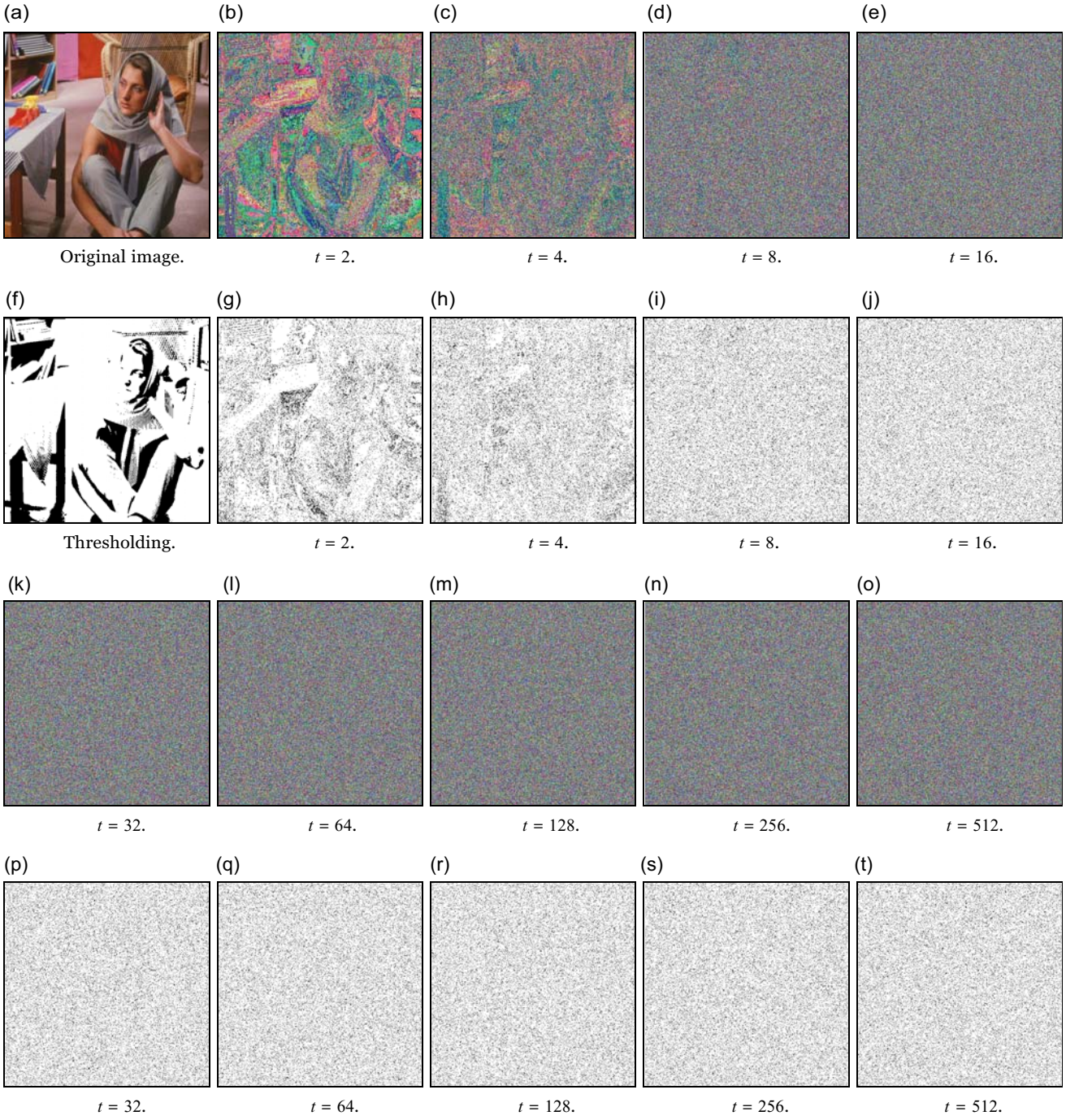
Figure 7 also illustrates the application of various edge detection techniques to an original image of a woman, serving as a visual exploration of the effects of different masks. In Figure 7(a), we have the unaltered original image. Subsequent Figures 7(b) through 7(i) also depict the results obtained by applying different edge detection filters. These filters, including the Laplace filter, isolated points, Roberts vertical and horizontal, Sobel horizontal and vertical, and Prewitt horizontal and vertical, emphasize distinct features within the image. Through variations in intensity and texture, the processed images unveil the presence of edges and contours, thereby contributing to a comprehensive analysis of the woman's image.

These filters are classical image processing techniques that enhance features, identify edges, and reduce noise, valuable in various applications from computer vision to medical imaging. This series of results helps understand the impact of diverse edge detection techniques on visual representation, providing insights for image processing and analysis.

3. Results and Discussion

The experiment conducted in this study involved the utilization of the three previously mentioned image segmentation techniques on digital images generated using the BCCA model. The images used in this experiment were created based on the application of transition CA-rule 30 at time steps $t = 2$ and $t = 8$, while the edge sequence employed by the BCCA was derived from an evolution of CA-rule 30 at $t = 512$ steps. The encryption was performed

Figure 9
Exemplification of image being encrypted using the BCCA method, and then, each encrypted image was thresholded

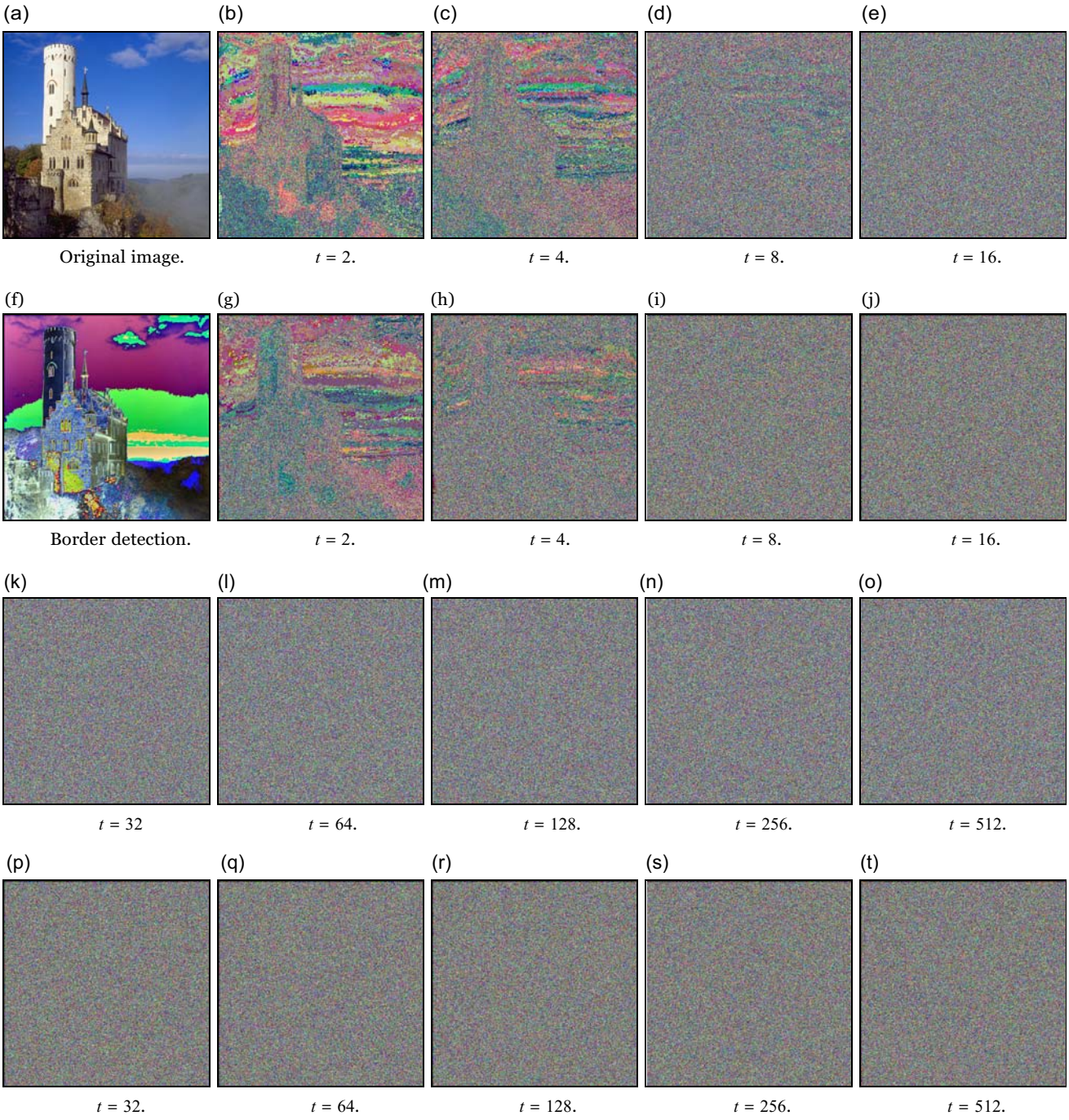


on a set of 12 images, each having dimensions of 512 512 pixels. Each of the segmentation techniques was individually applied to the resulting images. The primary objective of this experiment was to investigate whether segmentation techniques could serve as cryptanalysis procedures by highlighting specific characteristics or objects within images encoded at various time steps. By comparing the results across the spectrum of image qualities, the study aimed to determine whether the most securely encrypted images were susceptible to segmentation techniques.

3.1. Analyzing cipher steps using BCCA model

Firstly, a brute-force experiment was performed trying to identify the best amount of pre-image steps to be applied to the BCCA model. Thus, some of these ciphers are shown in Figure 8. We observe that as the size of the steps applied to the original image, see Figure 8(a), increases $T = 2$, 4, 8, 16, 32, 64, 256, 128, 512, see Figure 8(b)–8(j), the artifacts generated by the encryption decrease. However, a substantial increase in this parameter results in a higher number of encryption steps, which can degrade the

Figure 10
Exemplification of the image being encrypted using the BCCA method, and then, each image was segmented through the edge detection method

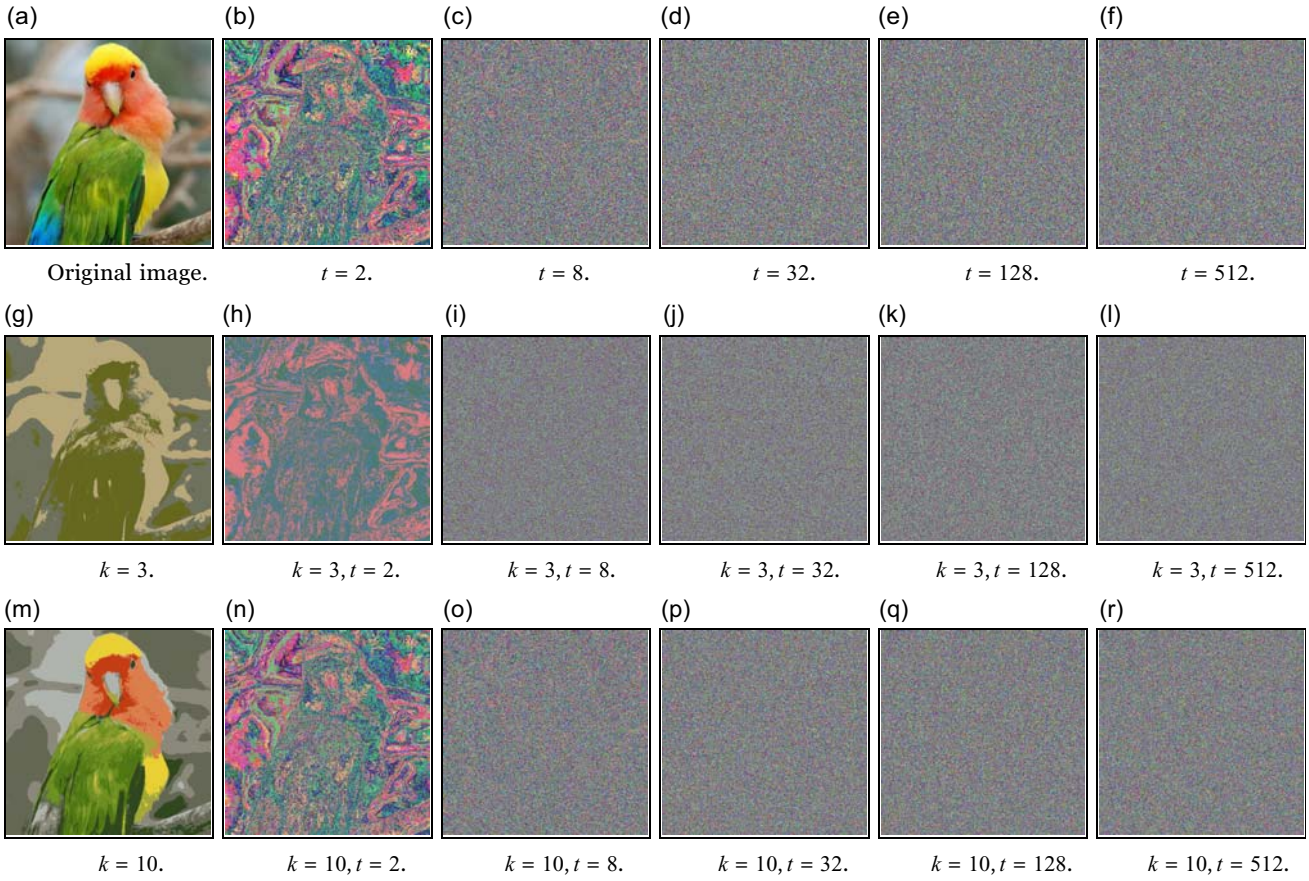


performance of the CA-based cryptographic model. Thus, it becomes essential to determine the optimal number of steps for achieving both effective encryption and efficient performance. Naturally, if we reduce the number of CA steps (*T)* for image encryption, the process will be faster. However, it is equally important to assess the image quality using segmentation techniques. Subsequent sections will introduce image segmentation-based cryptanalysis techniques aimed at enhancing the quality of the BCCA model's encryption.

3.2. Analyzing cipher by thresholding

The experiment involving the application of thresholding is illustrated in Figure 9. In Figure 9(a), the original image is depicted, and Figure 9(f) shows the same original image subjected to the thresholding technique. Moving forward, Figure 9(b) represents the image encoded at $t = 2$ steps using the BCCA cryptographic model, while Figure 9(g) presents the same image post-thresholding. For instance, Figure 9(d) is the result of

Figure 11
Exemplification of the image being encrypted using the BCCA method, and then, each encrypted image was grouped by the *k*-means algorithm



encryption at $t = 8$ steps and exhibits notable dissimilarities from its original version. However, the application of thresholding to this encrypted image, as seen in Figure 9(i), effectively reveals the presence of noise patterns that can be traced back to the original image. This underscores the insufficiency of employing a small number of steps for secure encryption. Figures 9(l) and 9(q) showcase an image encrypted at $t = 64$ steps, thus representing a more secure encryption compared to the preceding examples with fewer CA steps.

In another example, Figure 9(o) displays the original image encrypted using $t = 512$ pre-image calculation steps. However, when applying the thresholding technique to this image at $t = 512$, as shown in Figure 9(t), no discernible traces or noise patterns associated with the original image (Figure 9(a)) can be identified. This observation aligns with the initial analysis presented in the [12] article, indicating that the minimum number of pre-image steps T to be applied to the model is related to the larger dimension of one side of the image, denoted as $N = \max(m, n)$. This choice ensures the attainment of a distortion-free encryption, devoid of remnants or shadows from the original model, making it resilient against cryptanalysis attacks.

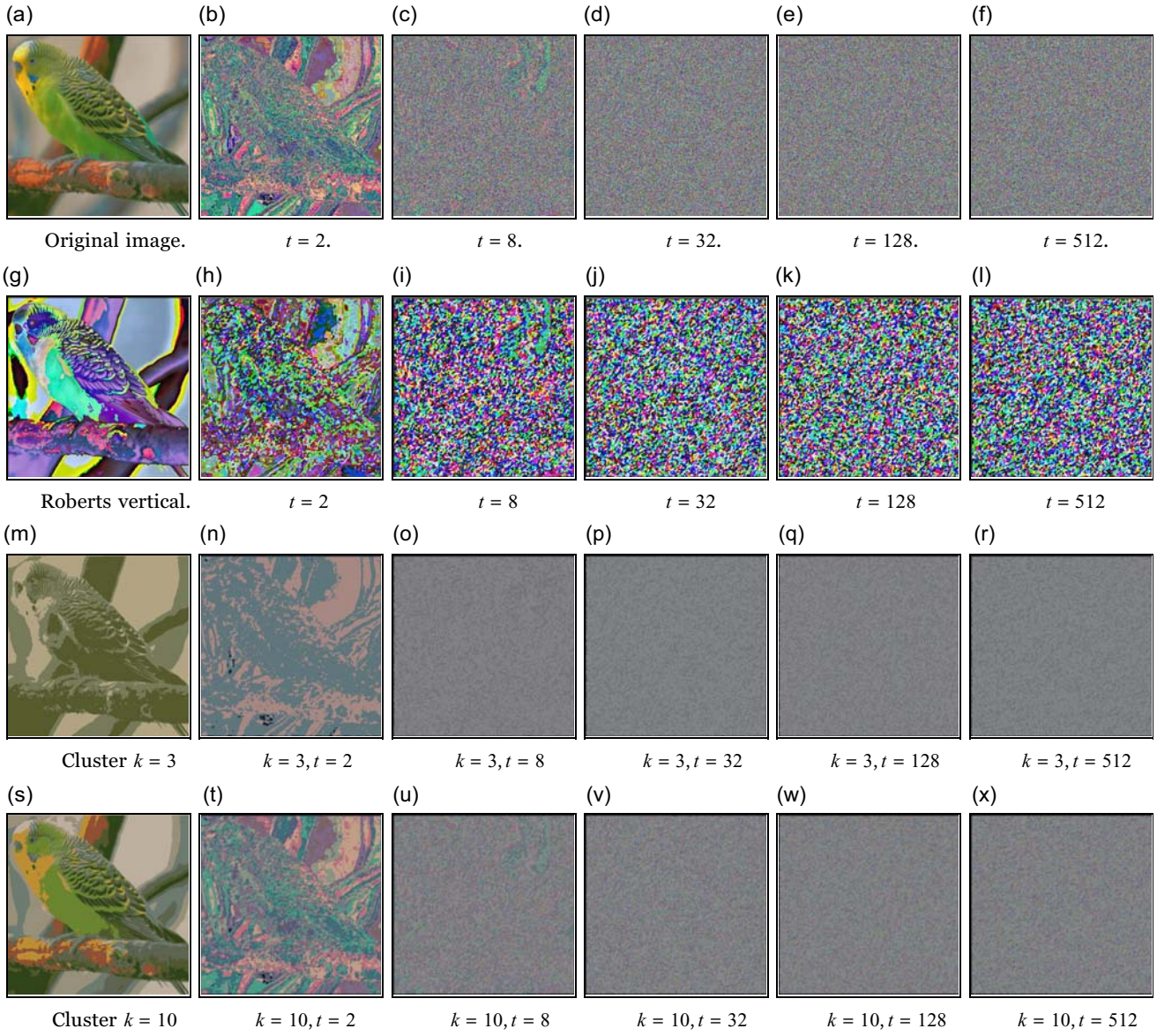
The encryption process introduces unique patterns into the image. Applying thresholding to these encrypted images produces the final set of figures. Thresholding distinguishes image regions based on intensity levels, revealing specific features. This series illustrates the profound impact of thresholding on BCCA-encrypted images, offering insights into their transformation and segmentation.

3.3. Analyzing cipher by edge detection method

Figure 10 provides a visual representation of the process involved in encrypting an image using the BCCA method and subsequently applying the edge detection technique to the encrypted images. It begins with the original image in Figure 10(a). The series continues by showcasing the image at different time steps ($t = 2$ to $t = 16$) post-encryption, as depicted in Figure 10(b) through 10(e) and Figure 10(g) through 10(j), respectively. CA steps ($t = 32$ to $t = 512$) are depicted in Figure 10(b) through 10(e), and the application of edge detection is shown in Figure 10(b) through 10(e) and Figure 10(g). The encryption process introduces alterations and patterns into the image. Subsequently, the edge detection technique is applied to each of these encrypted images, resulting in the final set of images presented in the figure.

Although Figure 10(c) presents a figure with little confidentiality, the application of the edge detection obtained little modification. Similarly, the figure in Figure 10(h) shows a small noise at the top. Edge detection was not effective in highlighting this disturbance. On the contrary, for images at later CA steps, such as the one in Figure 10(m), the edge detection technique, see Figure 10(r), was not able to highlight more prominent features,

Figure 12
Exemplification of the image being encrypted using the BCCA method, and then, each image filtered using the low pass filter, the images were later segmented using border detection and clustering



different from Figure 10(j) that remains an image noise after encryption shown in Figure 10(e). Varying edge detection effectiveness with different images and encryption steps highlights the need to choose an appropriate number of CA steps to balance confidentiality and image preservation. Border detection identifies edges and boundaries, facilitating region segmentation based on intensity or contrast variations. This image series demonstrates edge detection's impact on BCCA-encrypted images, revealing its role in transforming and segmenting visual content.

3.4. Analyzing cipher by clustering

Figure 11 demonstrates the process of encrypting an image using the BCCA method and then subjecting each encrypted original image, see Figure 11(a) to the *k*-means clustering algorithm. In the initial stages, the image goes through various encryption steps, represented by Figure 11(b) through 11(e), starting at $t = 2$ and progressing to $t = 512$. As the number of encryption steps increases, the visual characteristics of the image undergo transformation, resulting in varying levels of distortion and confidentiality. Subsequently, the *k*-means algorithm is applied to these encrypted images, with the parameter k indicating the number of clusters or groups into which the pixels in the image are categorized. This grouping process, as shown in Figure 11(g) ($k = 3$) through 11(r) ($k = 10$), aids in segmenting the image into distinct regions based on the similarity of pixel values. The experiment illustrates how the *k*-means clustering algorithm aids in evaluating encryption quality. With $T = 128$ steps, high-quality encryption is achieved, and using $k = 10$ in the *k*-means clustering algorithm improves noise analysis accuracy in the encrypted images. This pairing of encryption steps and clustering parameters enables a robust and precise assessment of encrypted image quality, facilitating the evaluation of any cipher remaining noise or artifacts.

3.5. Analyzing cipher using filters

This set of images, as depicted in Figure 12, illustrates the comprehensive analysis performed on an encrypted image using the BCCA method. It encompasses multiple stages of processing. In the context of edge detection, the vertical Roberts model [16] with the mask $\omega = [0,0,0,0,0,0,1,1]$ was used. The original image in Figure 12(a) is progressively encrypted using increasing time steps from $t = 2$ (Figure 12(b)) to $t = 512$ (Figure 12(f)).

Additionally, a vertical Roberts filter was applied to the encrypted images, which can be observed in Figure 12(g)–12(l). These filtered images are then clustered using two different values of k in the k-means algorithm, namely $k = 3$ (Figure 12(m)–12(r)) and $k = 10$ (Figure 12(s)–12(x)). Indeed, starting from $T = 128$, we can observe that the encryption is noise-free, validating the earlier findings from our previous experiments [12].

In contrast to methods like key avalanche analysis, the avalanche effect of encrypted images, entropy, and color histograms, segmentation techniques have provided the crucial validation of the requisite number of steps. This approach enables a comprehensive examination of how filtering and clustering affect the quality and attributes of encrypted images, providing invaluable insights for image processing and analysis within the framework of BCCA encryption. While it is important to combine this with the NIST framework, it is not the focus of this paper, as previously addressed in reference [12].

4. Conclusions

Images generated by the BCCA algorithm in a limited number of time steps may exhibit undesired artifacts that can compromise their confidentiality. To mitigate this issue, image segmentation techniques can serve as effective cryptanalysis tools for refining the proposed cryptographic model. For example, thresholding can accentuate image features, identify noise, and reconstruct objects in encrypted images. Clustering, on the other hand, can help consolidate pixels and remove the scrambling introduced by the cryptographic method. While edge detection yields favorable results with pristine original images, its performance may vary when applied to images with partial pixel scrambling.

The experiments have underscored the sensitivity of the encryption model to the number of steps involved in applying the set of rules. It is crucial to emphasize that a secure encryption process requires a minimum number of steps, ideally equivalent to the size of the largest dimension of the image. In our experiments, with square images, this minimum step count was necessary for reliable encryption. While increasing the number of steps could potentially enhance algorithm performance, especially on parallel hardware, it is evident from these cryptanalysis experiments that maintaining an adequate step count is essential. These findings highlight that robust encryption should not be vulnerable to segmentation-based cryptanalysis techniques, achievable by ensuring an ample number of steps in the process. Failure to adhere to this step requirement can render the BCCA model susceptible to cryptanalyst attacks.

Looking ahead, several potential future research directions emerge in this area. One promising avenue is the combination of various segmentation techniques to create new and more robust cryptanalysis methods. Additionally, exploring the integration of other common digital image processing techniques could facilitate the application of these segmentators and the subsequent analysis of the resulting images. For instance, incorporating filters or other types of pre-processing could potentially enhance the quality and effectiveness of the segmentation results and, consequently, the overall cryptanalysis process.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The data that support the findings of this study are openly available in SIPI Image Database at https://sipi.usc.edu/databas/database.php?volume=misc

Author Contribution Statement

Eduardo C. Silva: Software, Validation, Formal analysis, Investigation, Writing – original draft, Writing – review & editing. **Jaqueline N. Dorneles:** Software, Validation, Formal analysis, Investigation. **Danielli A. Lima:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration.

References

[1] Wang, X., Xue, W., & An, J. (2020). Image encryption algorithm based on tent-dynamics coupled map lattices and diffusion of household. *Chaos, Solitons & Fractals*, *141*, 110309.

[2] Dhanda, S. S., Singh, B., & Jindal, P. (2020). Lightweight cryptography: A solution to secure iot. *Wireless Personal Communications*, *112*, 1947–1980.

[3] Tedeschi, P., Sciancalepore, S., & Di Pietro, R. (2022). Satellite-based communications security: A survey of threats, solutions, and research challenges. *Computer Networks*, *216*, 109246.

[4] Borodzhieva, A. N. (2020). Computer-based tools applied in the course telecommunication security. In *Conference Proceedings of» eLearning and Software for Education «(eLSE)*, *16*, 42–52. Carol I National Defence University Publishing House.

[5] He, Y., Ye, N., & Zhang, R. (2021). Analysis of data encryption algorithms for telecommunication network-computer network communication security. *Wireless Communications and Mobile Computing*, *2021*, 1–19.

[6] Mittal, S., Bansal, A., Gupta, D., Juneja, S., Turabieh, H., Elarabawy, M. M., . . . , & Bitsue, Z. K. (2022). Using identity-based cryptography as a foundation for an effective and secure cloud model for e-health. *Computational Intelligence and Neuroscience*, *2022*(1), 7016554.

[7] Wan, S., Guan, S., & Tang, Y. (2023). Advancing bridge structural health monitoring: Insights into knowledge-driven and data-driven approaches. *Journal of Data Science and Intelligent Systems*, *2*(3), 129–140.

[8] Gunathilake, N. A., Al-Dubai, A., & Buchana, W. J. (2020). Recent advances and trends in lightweight cryptography for iot security. In *2020 16th International Conference on Network and Service Management*, 1–5.

[9] Noura, H., Chehab, A., Sleem, L., Noura, M., Couturier, R., & Mansour, M. M. (2018). One round cipher algorithm for multimedia iot devices. *Multimedia Tools and Applications*, *77*, 18383–18413.

[10] Sarveswaran, S., Shangkavi, G., Gowthaman, N., & Vasanthaseelan, S. (2021). Cryptography techniques and internet of things applications – A modern survey. *International Journal of Aquatic Science*, *12*(2), 2338–2371.

[11] Qowi, Z., & Hudallah, N. (2021). Combining caesar cipher and hill cipher in the generating encryption key on the vigenere cipher algorithm. *Journal of Physics: Conference Series*, *1918*, 042009.

[12] Silva, E. C., Soares, J. A., & Lima, D. A. (2016). Autômatos celulares unidimensionais caóticos com borda fixa aplicados à modelagem de um sistema criptográfico para imagens digitais. *Revista de Informática Teórica e Aplicada*, *23*(1), 250–276.

[13] Lira, E. R., de Macêdo, H. B., Lima, D. A., Alt, L., & Oliveira, G. M. (2023). A reversible system based on hybrid toggle radius-4 cellular automata and its application as a block cipher. *Natural Computing*, 1–17.

[14] Lopes, H. J., & Lima, D. A. (2022). Surveillance task optimized by evolutionary shared tabu inverted ant cellular automata model for swarm robotics navigation control. *Results in Control and Optimization*, *8*, 100141.

[15] Gutowitz, H. (1995). *Cryptography with dynamical systems*. UK: Kluwer Academic Press.

[16] Dorneles, J. N. (2018). *Analise de tecnicas de segmentacao em folhas de cafe*. Master's Thesis, Instituto Federal do Triangulo Mineiro Campus Patrocínio.

[17] Almuhammadi, S., & Alsaleh, M. (2017). Information security maturity model for nist cyber security framework. *Computer Science and Information Technology (CS & IT)*, *7*(3), 51–62.

[18] Calder, A. (2018). *NIST cybersecurity framework: A pocket guide*. UK: IT Governance Publishing Ltd.

[19] Khaleefah, A. D., & Al-Mashhadi, H. M. (2024). Methodologies, requirements, and challenges of cybersecurity frameworks: A review. *Iraqi Journal of Science*, *65*(1), 468–486.

[20] Krumay, B., Bernroider, E. W., & Walser, R. (2018). Evaluation of cybersecurity management controls and metrics of critical infrastructures: A literature review considering the NIST cybersecurity framework. In *Secure IT Systems: 23rd Nordic Conference, NordSec 2018, Proceedings 23*, 369–384.

[21] Shackelford, S. J., Proia, A. A., Martell, B., & Craig, A. N. (2015). Toward a global cybersecurity standard of care: Exploring the implications of the 2014 NIST cybersecurity framework on shaping reasonable national and international cybersecurity practices. *Texas International Law Journal*, *50*, 305.

[22] Gordon, L. A., Loeb, M. P., & Zhou, L. (2020). Integrating cost – benefit analysis into the nist cybersecurity framework via the gordon–loeb model. *Journal of Cybersecurity*, *6*(1), tyaa005.

[23] Wolfram, S. (2002). *A new kind of science*. Wolfram Media – (1st edition): 1197-2006-09-19T07:35:05.000+0200.

[24] Lima, M. J. L. (2005). *Criptografia baseada no calculo generico de pre-imagens de autômatos celulares*. Master's Thesis, Universidade Presbiteriana Mackenzie.

[25] Oliveira, G. M., Martins, L. G., Alt, L. S., & Ferreira, G. B. (2010). Exhaustive evaluation of radius 2 toggle rules for a variable-length cryptographic cellular automata-based model. In *Cellular Automata*, 275–286.

[26] Sen, S., Shaw, C., Chowdhuri, D. R., Ganguly, N., & Chaudhuri, P. P. (2002). Cellular automata based cryptosystem (CAC). In *Information and Communications Security,* 303–314.

[27] Shafique, A., Khan, K. H., Hazzazi, M. M., Bahkali, I., Bassfar, Z., & Rehman, M. U. (2023). Chaos and cellular automata-based substitution box and its application in cryptography. *Mathematics*, *11*(10), 2322.

[28] Stănică, G. C., & Anghelescu, P. (2024). Reversible cellular automata based cryptosystem. *Electronics*, *13*(13), 2515.

[29] Younes, O. S., Alharbi, A., Yasseen, A., Alshareef, F., Albalawi, F., & Albalawi, U. A. (2023). CeTrivium: A stream cipher based on cellular automata for securing real-timemultimedia transmission. *Computer Systems Science & Engineering*, *47*(3), 2895–2920

[30] Machicao, J., Marco, A. G., & Bruno, O. M. (2012). Chaotic encryption method based on life-like cellular automata. *Expert Systems with Applications*, *39*(16), 12626–12635.

[31] Anghelescu, P., Ionita, S., & Sofron, E. (2008). FPGA implementation of hybrid additive programmable cellular automata encryption algorithm. In *Hybrid Intelligent Systems, 2008. HIS'08. Eighth International Conference*, 96–101.

[32] Oliveira, G. M. B., Coelho, A., & Monteiro, L. (2004). Cellular automata cryptographic model based on bi-directional toggle rules. *International Journal of Modern Physics C*, *15*(08), 1061–1068

[33] Daemen, J., & Rijmen, V. (2005). Rijndael/aes. In H. C. A. van Tilborg (Ed.), *Encyclopedia of cryptography and security* (pp. 520–524). Springer.

[34] Hertz, L., & Schafer, R. W. (1988). Multilevel thresholding using edge matching. *Computer Vision, Graphics, and Image Processing*, *44*(3), 279–295.

How to Cite: Silva, E. C., Dorneles, J. N., & Lima, D. A. (2024). Qualitative Analysis of Cryptanalysis of Images Encrypted by Border Chaotic Cellular-Automaton-Based Model Using Segmentation Techniques. *Journal of Data Science and Intelligent Systems*. https://doi.org/10.47852/bonviewJDSIS42021976