Secret Communication Systems Using Chaotic Wave Equations with Neural Network Boundary Conditions

Chen, Yuhan
Sano, Hideki
Wakaiki, Masashi
Yaguchi, Takaharu

(Citation)
Entropy,23(7):904

(Issue Date)
2021-07

(Resource Type)
journal article

(Version)
Version of Record

(URL)
https://hdl.handle.net/20.500.14094/90008524

entropy

MDPI

Article

Secret Communication Systems Using Chaotic Wave Equations with Neural Network Boundary Conditions

Yuhan Chen *, Hideki Sano, Masashi Wakaiki and Takaharu Yaguchi

Graduate School of System Informatics, Kobe University, Kobe 657-8501, Japan; sano@crystal.kobe-u.ac.jp (H.S.); wakaiki@ruby.kobe-u.ac.jp (M.W.); yaguchi@pearl.kobe-u.ac.jp (T.Y.)
* Correspondence: 193x226x@stu.kobe-u.ac.jp

Abstract: In a secret communication system using chaotic synchronization, the communication information is embedded in a signal that behaves as chaos and is sent to the receiver to retrieve the information. In a previous study, a chaotic synchronous system was developed by integrating the wave equation with the van der Pol boundary condition, of which the number of the parameters are only three, which is not enough for security. In this study, we replace the nonlinear boundary condition with an artificial neural network, thereby making the transmitted information difficult to leak. The neural network is divided into two parts; the first half is used as the left boundary condition of the wave equation and the second half is used as that on the right boundary, thus replacing the original nonlinear boundary condition. We also show the results for both monochrome and color images and evaluate the security performance. In particular, it is shown that the encrypted images are almost identical regardless of the input images. The learning performance of the neural network is also investigated. The calculated Lyapunov exponent shows that the learned neural network causes some chaotic vibration effect. The information in the original image is completely invisible when viewed through the image obtained after being concealed by the proposed system. Some security tests are also performed. The proposed method is designed in such a way that the transmitted images are encrypted into almost identical images of waves, thereby preventing the retrieval of information from the original image. The numerical results show that the encrypted images are certainly almost identical, which supports the security of the proposed method. Some security tests are also performed. The proposed method is designed in such a way that the transmitted images are encrypted into almost identical images of waves, thereby preventing the retrieval of information from the original image. The numerical results show that the encrypted images are certainly almost identical, which supports the security of the proposed method.

Keywords: chaotic synchronization; secret communication system; van der Pol boundary condition; deep learning

Citation: Chen, Y.; Sano, H.; Wakaiki, M.; Yaguchi, T. Secret Communication Systems Using Chaotic Wave Equations with Neural Network Boundary Conditions. *Entropy* **2021**, *23*, 904. https://doi.org/10.3390/e23070904

Academic Editors: José A. Tenreiro Machado, Carla M.A. Pinto, Julio Rebelo and Helena Reis

Received: 31 May 2021
Accepted: 12 July 2021
Published: 16 July 2021

1. Introduction

Innovations in information processing technology have led to the rapid development of technologies for accessing and transmitting information in recent years, which has greatly improved the convenience of our daily lives. On the other hand, a wide variety of information is exchanged in public through the Internet and other media, so encryption technology has become increasingly important to protect this information.

In this paper, we consider confidential communication methods using chaos synchronization as one of the encryption techniques. Applying the diversity of chaotic series, pseudo-random series can be generated with statistical properties close to those of ideal random series, which can hide the original information. The problem of designing confidential communication systems using chaotic synchronization has been studied since the 1990s. The core idea of the method is to embed communication information into a signal that behaves chaotically. The method accomplishes the steganography and the recovery of

communication information [1–5]. For example, by applying the Hénon mapping, a chaotic synchronous control method for discrete-time nonlinear systems can be designed with significant results [4]. A Hénon map (Hénon–Pomeau attractor/map) is a discrete-time dynamical system that can generate chaotic phenomena, with an iterative expression of:

$$\begin{cases} x_{n+1} = 1 - ax_n^2 + y_n \\ y_{n+1} = bx_n, \end{cases} \tag{1}$$

where a, b are parameter values. When they are set to appropriate values, the system will exhibit chaotic behaviors. In [6], the color image encryption algorithms applying scrambling-diffusion are improved by introducing the transforming-scrambling-diffusing model so that the methods have better security and cryptography characteristics. In [7], a SystemC implementation of a chaos-based crypto-processor for the AES algorithm is presented, where the properties of chaotic systems are employed to cope with the parameters of the AES algorithm. An image encryption algorithm is proposed in [8], where the Arnold chaos sequence and the modified AES algorithm are combined. In [9], a method to encrypt multiple images is proposed by a combination of a fast chaotic encryption algorithm and the AES algorithm. A family of complex variable chaotic systems are used to develop an image encryption algorithm in [10]. In [11], hyper-chaos systems and DNA sequences are combined for encrypting images, where the pseudo-random sequence generated by a hyper-chaos system is transformed into a DNA sequence to diffuse the image blocks.

However, in most of the above methods, the information on the chaotic systems must be shared for encryption and decryption to generate the chaotic sequences. In other words, the parameters of the system and the initial conditions for defining the state of the system essentially played the role of the keys. Although the algorithms are quite secure under the assumption that the information on the chaotic system can be shared safely, there remains a risk of information leakage if this is not the case. Several papers have proposed methods to address this problem by using chaotic synchronization [3,12–17]. The chaotic systems have positive Lyapunov exponents, which means that small differences in the initial conditions will grow exponentially. Therefore, the phenomenon of synchronization of such systems is surprising; however, it is known that such synchronization can actually occur [18] and much attention has been focused on this phenomenon. If the chaotic systems of the receiver and the sender can be synchronized by introducing appropriate control signals, the sharing of the parameters and the initial conditions becomes unnecessary. In particular, in [3], chaos-based synchronized dynamic keys are designed and an improved chaos-based advanced encryption standard (AES) algorithm is developed so that a powerful method is proposed that solves the problem of using a static key for AES, while retaining the advantages of the chaos synchronization-based method.

We employ a different approach to these methods, in which, while the problem of having to share the initial conditions is addressed by the chaos synchronization technique, that of having to share the parameters of the systems is overcome by introducing extremely complex parameterization of the chaotic systems, that is, neural networks. More precisely, we use certain chaotic phenomena in distributed systems with nonlinear boundary conditions represented by a certain neural network. The neural network is pre-trained so that it defines a chaotic function. As far as the networks are chaotic, neural networks with arbitrary architectures can be used. Therefore, the number of the neural networks, and hence the size of the key space, can be arbitrarily large by enlarging the network architecture.

The construction of a confidential communication system using chaotic phenomena in distributed systems has also been studied before but without the neural-network boundary condition. In this paper, we focus on the system proposed in [19], in which a certain initial boundary value problem of the linear wave equation is employed on a one-dimensional bounded interval, with a linear homogeneous boundary condition at the left end and the nonlinear boundary condition, which has a cubic nonlinearity of the van der Pol

type [20,21]. The interaction of these linear and nonlinear boundary conditions leads to chaos in the Riemann invariants (u, v) of the wave equation when the parameters satisfy certain requirements. By constructing an observer by applying the method of characteristics, the appropriate range of the feedback gain is obtained so that the convergence of the dynamics that describes the synchronization error between two mappings is ensured. Through the numerical computation, it is also confirmed that the one-dimensional wave equation with the van der Pol type boundary conditions exhibits a spatio-temporal chaotic behavior in its dynamics [22]. In [19], this chaotic vibration of the wave equation under the van der Pol boundary condition is specifically used to construct the synchronous system. There are two features of this approach—firstly, the synchronization system is easy to construct and secondly, it transmits vector-valued signals in a secure communication system [19].

On the other hand, many information security techniques are based on artificial intelligence. As the problem of image recognition is a specialty of neural networks, neural networks are often used in information security technologies to solve problems of image recognition and image analysis. Biometric technologies, such as face recognition systems and fingerprint authentication, are examples of this [23]. Meanwhile, in terms of security performance, deep learning also does not have the complete capability to guarantee absolute security and privacy. There are many types of attacks that are made on deep learning, such as Causative Attacks, Exploratory Attacks and Indiscriminate Attacks [24]. With these attacks, the model information or the knowledge of the training data can be extracted; these are known as model inversion, model extraction and membership inference, where the attackers steal the training data and produce the expected results, or provide incorrect training data. This means that the attackers may have the ability to change the inputs to the training data, which becomes the reason for parameter changes in the learning model, leading to a significant decrease in the performance of the subsequent classification tasks. To address these leakage risks, privacy-preserving learning, such as Defensive Distillation, has been developed for defending against poisoning attacks; however, these approaches cannot eliminate all security risks [24] at this time.

In this paper, we combine chaos theory and deep learning to construct a model that makes it difficult to steal information and that has a stronger secrecy effect, and apply it to a confidential communication system for color images.

This research aims to provide an information communication system with high secrecy performance, which uses the wave equation with chaotic behaviors and also deep neural networks. More precisely, the proposed approach is to apply a synchronous pair of chaotic distribution systems to encrypt the transmitted object, while applying a deep neural network as a black box to encrypt and decrypt color image information. The following is a brief description of the proposed approach.

Firstly, the proposed approach employs the chaotic wave equation, which is an evolutional partial differential equation and hence has two axes (space and time). This feature of the equation enables the method to transmit images. Secondly, the proposed secret communication system consists of two parts, the sender and the receiver, as shown in Figure 1. First, the wave equation on the sender side is under the van der Pol boundary conditions that can cause chaotic phenomena, and a deep neural network is trained so that it has the same chaotic effect. Then, the color images are encrypted by applying a certain nonlinear transformation together with the solutions to the wave equations. The nonlinear transformation is designed so that the inverse of it is computable if the solutions to the wave equations, which are essentially the secret keys, are known. Second, a chaotic synchronization system is established at the receiver side to obtain the decrypted recovered images by inverting the function.

The innovation of this study is to use a deep neural network that approximates the boundary conditions yielding the chaotic vibrations, which addresses the problem that the previous system is easier to be cracked and makes it easier for the information to be stolen due to too few parameters of the van der Pol boundary condition when the original chaotic

synchronization system is used alone. In fact, in the wave equation with the van der Pol boundary conditions, the three parameters required are a, β and η (see Section 2 for the specific expressions). Suppose that a hacker now steals the specific value of β, and the other two (a and η) still exist in a hidden state. Suppose also that that person makes an effort to guess the values of a and η and tries to break the encrypted information. In Figure 2, we can see that the possibility exists that the whole information of the portrait can be seen to some extent. Although it was mentioned in the above that deep learning techniques are also under threat from security attacks, a stealer will face greater implementation difficulties than a brute force attack when parameters are partially compromised. More notably, the way the proposed approach corresponds the neural network structure with the boundary conditions at the left end and the right end will also increase the difficulty for the stealer to crack.

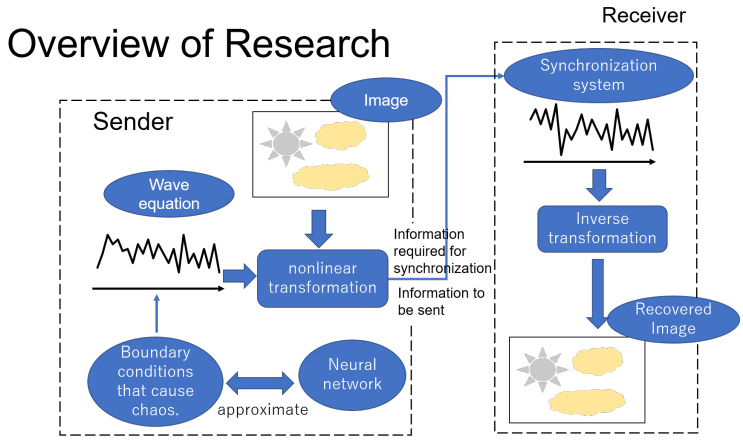

Figure 1. A diagrammatic representation of the research ideas and methods used in this study. It succinctly shows the overall structure of the confidential communication system as well as the design principles, in which the input portrait was created manually by the author and is only used here as an example to assist in illustration.

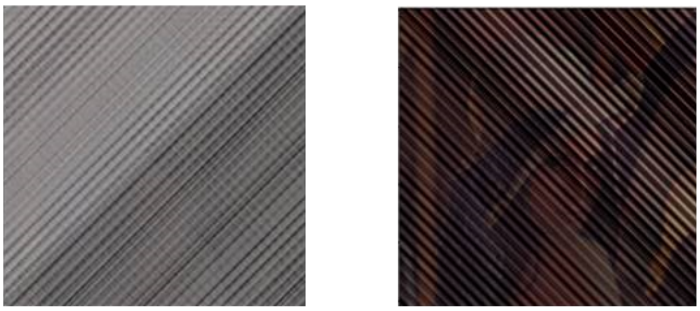

Figure 2. The modulated secrecy image and the restored image after the leakage of some of the parameters of the confidential communication system. The figure on the left is the confidential information after it has been encrypted, and we can see that we have no way of knowing the information about the original image, so we can say that it was a successful secret communication. The figure on the right is the restored image after the leakage of some of the parameters of the confidential communication system; we can almost see that it is a picture of a woman wearing a hat, that is to say, the hacker can restore the transmitted information to some extent after breaking some of the parameters.

This paper is organized as follows. First, in Section 2, the proposed method for grayscale images is described in detail along with some numerical examples. In Section 3, the method is applied to color images. In Section 4, the encryption effects of the proposed

method and the AES are tested, respectively, and the results are analyzed. Concluding remarks are shown in Section 5.

2. Grayscale Images as Transmission Objects

Before talking about the system for color images, we will explain the distributed system with chaotic vibrations and synchronization systems using grayscale images as transmitted objects, and numerically test it to obtain the encoded images and the restored images, and compare them with the original images, thereby investigating the feasibility of the proposed approach. Since grayscale images do not involve RGB, the system becomes simpler and hence it is somewhat easier to understand than with color images.

2.1. Wave Equation with the van der Pol Boundary Conditions

The system we consider here is a linear PDE but with a nonlinear boundary condition, which is from the van der Pol equation without forcing,

$$\begin{cases} \ddot{x} + (-\alpha\dot{x} + \beta\dot{x}^3) + \omega_0^2 x = 0, \quad \alpha, \beta, \omega^2 > 0 \\ x(0) = x_0, \quad \dot{x}(0) = x_1, \quad x_0, x_1 \in \mathbb{R}, \end{cases} \tag{2}$$

where ω_0 is the fixed frequency of the corresponding linear harmonic oscillator. The energy of this system is given by

$$E(t) = \frac{1}{2}[\dot{x}(t)^2 + \omega_0^2 x(t)^2], \tag{3}$$

and the time rate of change of energy is

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}E(t) &= \dot{x}(t)[\ddot{x}(t) + \omega_0^2 x(t)] \\ &= \alpha\dot{x}(t)^2 - \beta\dot{x}^4(t), \end{aligned} \tag{4}$$

so we get

$$\begin{cases} \geq 0, & \text{if} \quad |\dot{x}| \leq \left(\frac{\alpha}{\beta}\right)^{\frac{1}{2}} \\ \leq 0, & \text{if} \quad |\dot{x}| > \left(\frac{\alpha}{\beta}\right)^{\frac{1}{2}}, \end{cases} \tag{5}$$

which shows the self-regulation effect, as the energy will increase when $|\dot{x}|$ is small, and the energy will decrease when $|\dot{x}|$ is large. Therefore, unless the initial condition satisfies $x_0 = x_1 = 0$ in (2), causing $E(t) = 0$ for all $t > 0$, we can know that $E(t)$ will rise and fall between a certain interval of the (B_1, B_2). The bounds B_1 and B_2 can be determined by the parameters α and β.

Let us describe a wave equation below as

$$\frac{1}{c^2}y_{tt}(x,t) - y_{xx}(x,t) = 0, \qquad 0 < x < 1, \quad t > 0, \tag{6}$$

which defines the linear PDE that describes a vibrating string on the unit interval $(0,1)$, where $c > 0$ denotes the speed of wave propagation. Because the speed of wave c is not an essential parameter, we set $c = 1$. Thus, in this paper, we consider

$$y_{tt}(x,t) - y_{xx}(x,t) = 0, \qquad 0 < x < 1, \quad t > 0. \tag{7}$$

At the left-end $x = 0$, we choose the following boundary condition:

$$y_t(0,t) = -\eta y_x(0,t), \qquad t > 0, \quad \eta > 0, \quad \eta \neq 1. \tag{8}$$

At the right-end $x = 1$, we assume a nonlinear boundary condition,

$$y_x(1,t) = \alpha y_t(1,t) - \beta y_t^3(1,t), \qquad t > 0, \quad \alpha, \beta > 0. \tag{9}$$

The initial conditions are:

$$y(x,0) = y_0(x), \quad y_t(x,0) = y_1(x), \qquad 0 \le x \le 1 \tag{10}$$

for two given smooth functions y_0 and y_1.

As a summary, the wave equation with the van der Pol boundary condition is given by

$$\begin{cases} y_{tt}(x,t) - y_{xx}(x,t) = 0, \quad 0 < x < 1, \quad t > 0 \\ y_x(1,t) = \alpha y_t(1,t) - \beta y_t^3(1,t), \quad t > 0 \\ y_t(0,t) = -\eta y_x(0,t), \quad t > 0 \\ y(x,0) = y_0(x), \quad y_t(x,0) = y_1(x), \quad 0 \le x \le 1, \end{cases} \tag{11}$$

where we set $\alpha, \beta, \eta > 0$ and $\eta \neq 1$. We use the method of characteristics to rewrite (11). Let u and v be the Riemann invariants [25] of (11)

$$\begin{cases} u(x,t) = \frac{1}{2}\{y_x(x,t) + y_t(x,t)\} \\ v(x,t) = \frac{1}{2}\{y_x(x,t) - y_t(x,t)\}, \end{cases} \tag{12}$$

with initial conditions

$$\begin{cases} u(x,0) = u_0(x) = \frac{1}{2}[y_0'(x) + y_1(x)] \\ v(x,0) = v_0(x) = \frac{1}{2}[y_0'(x) - y_1(x)] \end{cases} \quad 0 \le x \le 1. \tag{13}$$

Then, the substitution of u and v into the boundary condition (9) gives

$$u(1,t) = F_{\alpha,\beta}(v(1,t)), \quad t > 0. \tag{14}$$

Besides, since $y_x(1,t) = u(1,t) + v(1,t)$, $y_t(1,t) = u(1,t) - v(1,t)$ at the right end $x = 1$, the relationship (14) becomes

$$\beta(u-v)^3 + (1-\alpha)(u-v) + 2v = 0. \tag{15}$$

Thus, let us summarize what the 1st order hyperbolic equation of (13) looks like after using Riemann invariants u and v.

$$\Sigma_0 : \begin{cases} u_t(x,t) = u_x(x,t), \quad x \in (0,1), \quad t > 0 \\ v_t(x,t) = -v_x(x,t), \quad x \in (0,1), \quad t > 0 \\ u(1,t) = F_{\alpha,\beta}(v(1,t)), \quad t > 0 \\ v(0,t) = qu(0,t), \quad t > 0 \\ u(x,0) = \frac{1}{2}[y_0'(x) + y_1(x)] = u_0(x), \quad x \in [0,1] \\ v(x,0) = \frac{1}{2}[y_0'(x) - y_1(x)] = v_0(x), \quad x \in [0,1], \end{cases} \tag{16}$$

where $q = \frac{1+\eta}{1-\eta}$ by (12), from which follows

$$v(0,t) = q(u(0,t)) = \frac{1+\eta}{1-\eta} u(0,t), \quad t > 0. \tag{17}$$

(15) and (17) represent the boundary conditions of the two boundaries at which the waves are reflected, respectively, as shown in Figure 3. When the wave reaches the right end $x = 1$, it is reflected to the left via (15) changing direction; when it reaches the left end, it is transmitted to the right via (17).

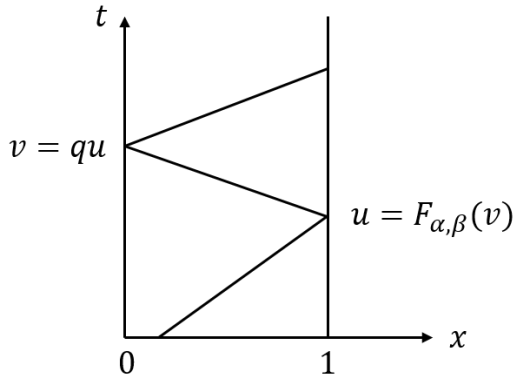

Figure 3. Reflection of the characteristic lines. The waves are reflected at $x = 0$ and $x = 1$, where the two functions $v = qu$ and $u = F_{\alpha,\beta}(v)$ are applied.

For (15), with $0 < \alpha \leq 1$, there exists a unique $u \in \mathbb{R}$ corresponding to each $v \in \mathbb{R}$ [20]. On the other hand, when $\alpha > 1$, for each $v \in \mathbb{R}$, in general there exist two or three distinct $u \in \mathbb{R}$, satisfying (15). Thus, in the latter case, $u = F_{\alpha,\beta}(v)$ is not well-defined, and hence the original PDE system (11) is not unique.

For parameters, for example, $\alpha = 0.5, \beta = 1$ and $\eta = 0.625$ are used, but special attention should be paid to the setting of η, which has theoretically been shown to exponentially increase the total variation of the system if it is chosen as an appropriate value. Since total variation is one of the indicators of the severity of the function change, its increase means that the system behaves chaotically.

So far, we have obtained a system of PDEs, which is a wave equation that can cause chaotic vibrations. This system allows us to chaoticize the transmitted image, and in order to restore the chaotic image in a later step, we need to construct a synchronization system. This system is designed so that the system exhibits exactly the same chaotic vibrations of the original system. In the next section, we will construct such a synchronization system.

2.2. Synchronization System

The construction of a synchronization system requires a portion of the original system's information. Hence, the sender sends two signals to the receiver: one is a secret, coded image, and the other is the signal needed for the synchronized system to restore the original system's state. For the system $\sum_0$ (16), consider the following system $\sum_1$:

$$\Sigma_1 : \begin{cases} \hat{u}_t(x,t) = \hat{u}_x(x,t), \quad x \in (0,1), \quad t > 0 \\ \hat{v}_t(x,t) = -\hat{v}_x(x,t), \quad x \in (0,1), \quad t > 0 \\ \hat{u}(1,t) = F_{\alpha,\beta}(v(1,t)), \quad t > 0 \\ \hat{v}(0,t) = qu(0,t), \quad t > 0 \\ \hat{u}(x,0) = \hat{u}_0(x), \quad x \in [0,1] \\ \hat{v}(x,0) = \hat{v}_0(x), \quad x \in [0,1], \end{cases} \tag{18}$$

with two signals, $u(0,t), v(1,t)$, as inputs. The relationship between the two systems is shown in Figure 4. We set variables $\widetilde{u} = u - \hat{u}$, $\widetilde{v} = v - \hat{v}$ to obtain the error system as follows:

$$\begin{cases} \widetilde{u}_t(x,t) = \widetilde{u}_x(x,t), \quad x \in (0,1), \quad t > 0 \\ \widetilde{v}_t(x,t) = -\widetilde{v}_x(x,t), \quad x \in (0,1), \quad t > 0 \\ \widetilde{u}(1,t) = 0, \quad t > 0 \\ \widetilde{v}(0,t) = 0, \quad t > 0 \\ \widetilde{u}(x,0) = \widetilde{u}_0(x), \quad x \in [0,1] \\ \widetilde{v}(x,0) = \widetilde{v}_0(x), \quad x \in [0,1]. \end{cases} \tag{19}$$

Solving the system using the method of characteristics, we can see that, at any initial value $\widetilde{u}_0$, $\widetilde{v}_0$, the solution $\widetilde{u}(\cdot,t)$ and $\widetilde{v}(\cdot,t)$ is completely zero at moment $t=2$. In other words, at moment $t=0$, there is no error between system (18) and system (16) and they reach a synchronized state.

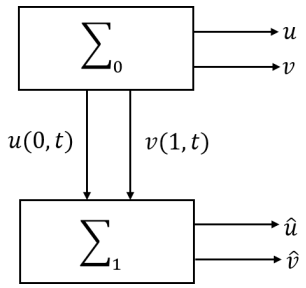

Figure 4. Synchronization system $\sum_1$.

In the previous study [19], it was assumed that the parameters of the system and information about the boundary conditions necessary to define (18) are shared in advance. The sender then sends the modulated information using the states of the chaotic system for secret communication, while the receiver sends the values $u(0,t)$ and $v(1,t)$ at the boundary required for synchronization at the sender to the synchronization system to evolve and demodulate the information by using the system state used during modulation.

2.3. Proposed Secret Communication System

In this study, we use neural networks to improve the security of the method by black-boxing the boundary conditions. Since the experimental objects in this section are grayscale images, the images can be transmitted as a single signal through the secure communication system. In the paper [19], in particular, the modulation and demodulation of the $(M+1)\times(L+1)$ pixel image data are studied by extending the method of the paper [4] to the distribution system. For this purpose, system (16) and synchronization system (18) must be discretized in both spatial and temporal directions.

Divide the interval $[0,1]$ equally into L parts and set the division points $x_i = i\Delta x (i=0,1,\dots,L)$. Here, the interval is given by $\Delta x = \frac{1}{L}$. The time step size is set to Δt and we write $t_k = k\Delta t (k=0,1,\dots)$. For the system (16) $u(x,t), v(x,t)$ denote the states, $u_i(k), v_i(k)$ denote the approximation values of $u(x_i,t_k), v(x_i,t_k)$ on the grid points (x_i,t_k). For simplicity, we introduce the $(L+1)$ dimension vectors $u(k)=[u_0(k),u_1(k),\dots,u_L(k)]^T, v(k)=[v_0(k),v_1(k),\dots,v_L(k)]^T$. Denote $\hat{u}_i(k)=\hat{u}(x_i,t_k)$, $\hat{v}_i=\hat{v}(x_i,t_k)$ for system (18) as well, and denote the $(L+1)$ dimension vector $\hat{u}(k)=[\hat{u}_0(k)$, $\hat{u}_1(k),\dots,\hat{u}_L(k)]^T$, $\hat{v}(k)=[\hat{v}_0(k),\hat{v}_1(k),\dots,\hat{v}_L(k)]^T$. Thus, for example, in Figure 4, the signals $u(0,t), v(1,t)$ sent from $\sum_0$ to $\sum_1$ for synchronization are discretized to $u_0(k), v_L(k)$, respectively. The time evolution of the vector $u(k), v(k)$ is shown in Figure 5. We approximate the $\sum_0$ and $\sum_1$ by the upwind difference using the same step sizes in the spatial and temporal directions so that the CFL number is 1. As a result, we obtain an algorithm where $u_i(k)$ is transported to $u_{i-1}(k+1)$ and $v_{i-1}(k)$ is to $v_i(k+1)$. More precisely, the system after discretization is

$$\begin{cases} \frac{u_i(k+1)-u_i(k)}{\Delta t} = \frac{u_{i+1}(k)-u_i(k)}{\Delta x}, & k>0, i\in\{1,2,\dots,L-1\} \\ \frac{v_i(k+1)-v_i(k)}{\Delta t} = -\frac{v_i(k)-v_{i-1}(k)}{\Delta x}, & k>0, i\in\{1,2,\dots,L-1\} \\ u_L(k) = F_{\alpha,\beta}(v_L(k)), & k>0, \\ v_0(k) = qu_0(k), & k>0. \end{cases} \tag{20}$$

Here, we set $\Delta t = \Delta x = \frac{1}{L}$ as the time and space step sizes, which gives

$$\begin{aligned} u_i(k+1) &= u_{i+1}(k), \quad k>0, i\in\{1,2,\dots,L-1\} \\ v_i(k+1) &= v_{i-1}(k), \quad k>0, i\in\{1,2,\dots,L-1\}. \end{aligned} \tag{21}$$

$\sum_1$ is discretized in the same manner. Henceforth, the systems $\sum_i (i = 0, 1)$ after discretization are denoted by $\overline{\sum}_i (i = 0, 1)$.

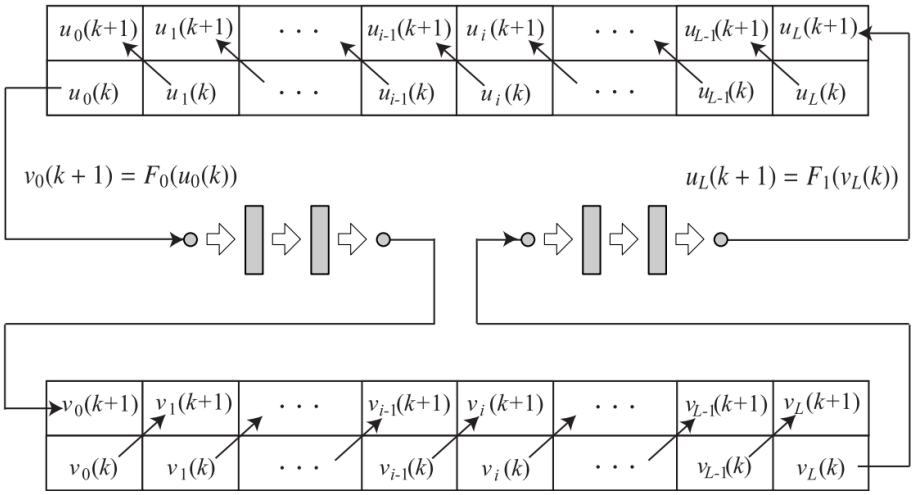

Figure 5. Time evolution of $u(k)$ and $v(k)$ in the discretized systems with the boundary conditions given by the neural networks.

As we mentioned earlier, due to the risk of theft due to the small number of parameters of the wave Equation (16) with the van der Pol boundary condition, in this study we propose a way to enhance security by approximating the van der Pol boundary condition with a neural network instead of the original boundary condition. The first half of this section describes how the u, v evolve over time, and the two systems $\sum_0, \sum_1$ are discretized. As shown in Figure 5, in the previous method, the waves at the boundaries are updated by $v_0(k+1) = qu_0(k)$ and $u_L(k+1) = F_{\alpha,\beta}(v_L(k))$, respectively. In the proposed method, which uses a neural network instead of these boundary conditions, we need two functions, F_1 and F_0, in order to simulate the boundary conditions.

One of the properties about neural networks is that they can approximate any continuous functions defined on compact sets. That is, neural networks can be a complicated, wiggly function, $f(x)$, as in Figure 6.

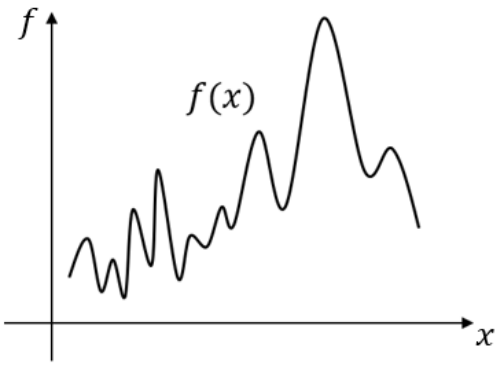

Figure 6. A complex continuous function $f(x)$.

For every possible input x, no matter what function it is, there is a neural network whose output value is close to $f(x)$. This property holds for functions with multiple inputs $f = f(x_1, ..., x_m)$ and multiple outputs. This property of neural networks is called the universal approximation property. Moreover, this universality theorem holds even for neural networks that have only one hidden layer between the input and output layers. In other words, the expressive power is extremely high even for extremely simple network structures.

The neural network applied here is composed of seven layers of perceptrons, as shown in Figure 7, and is approximated to $qF_{\alpha,\beta}$ by training. Since in this section we suppose that the images are of grayscale, both the input and output layers of the neural network have a single neuron. In this proposed approach, the fourth layer is also designed as a single neuron. The whole neural network is divided into two parts; the first part is the first to fourth layers, which is assumed to represent a function F_0, and the second part is the fifth to seventh layers, which is assumed to represent a function F_1. As shown in Figure 5, using the learned neural network, the left boundary condition is replaced, from $v_0(k+1) = qu_0(k)$ to the front half of the neural network F_0, while the right boundary condition is replaced, from $u_L(k+1) = F_{\alpha,\beta}(v_L(k))$ to the back half of the neural network F_1. In this way, the boundary conditions are black-boxed and the information becomes difficult to leak.

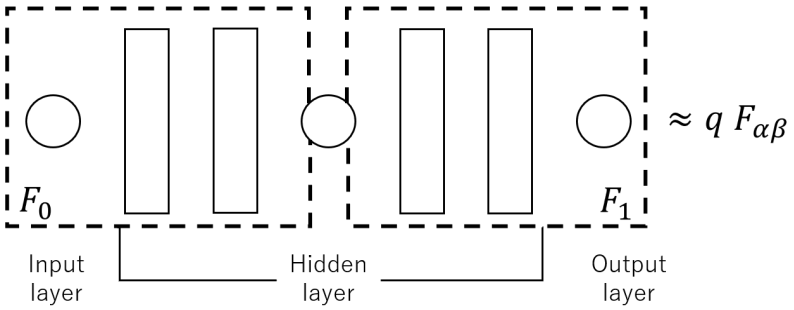

Figure 7. Relationship between the structure of neural networks and boundary conditions.

After approximating the van der Pol boundary condition, we tested the resulting new neural network by computing its Lyapunov exponent. In fact, when dealing with actual chaotic phenomena, instead of providing a clear mathematical definition of chaos, whether the system is chaotic or not is checked by a practical condition, which is determined by a certain criterion that is numerically computable. A dynamical system F is chaotic if it satisfies at least one of the following conditions:

(1) F has a sensitive dependence on initial conditions within the defined region.
(2) F has positive Lyapunov exponents at all points in the definite domain excluding the final immobile point [26,27].

In this study, we use the method of calculating the Lyapunov exponents, that is, condition (2). Lyapunov exponents are used to quantify the separation rate between infinitely close trajectories in a dynamical system. Specifically, under the assumption that linearization is feasible, the separation rate of the two trajectories with an initial interval of σZ_0 is

$$|\sigma Z(t)| \approx \mathrm{e}^{\lambda t}|\sigma Z_0|, \tag{22}$$

where λ is the Lyapunov exponents. In this section, where the images are assumed to be grayscale, since no color issues are involved, it can be viewed as a one-dimensional discrete-time system, in which case the Lyapunov exponent λ for a one-dimensional map $x_{n+1} = L(x_n)$ is defined by

$$\lambda = \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N-1} \log |L'(x_i)|, \tag{23}$$

where $x \in \mathbb{R}$ is the state variable of the system and $n \in \mathbb{N}$ is the discrete time. If the value of the Lyapunov exponent computed in (23) is positive, then the system F is considered to be chaotic. Therefore, we use this method to judge whether the trained neural network successfully approximates the chaotic functions.

As shown in Figure 8, the specific components of the whole secret communication system are the system $\overline{\sum}_0$, synchronization system $\overline{\sum}_1$, modulation M and demodulation D, where modulation and demodulation, respectively, are represented by the following equations:

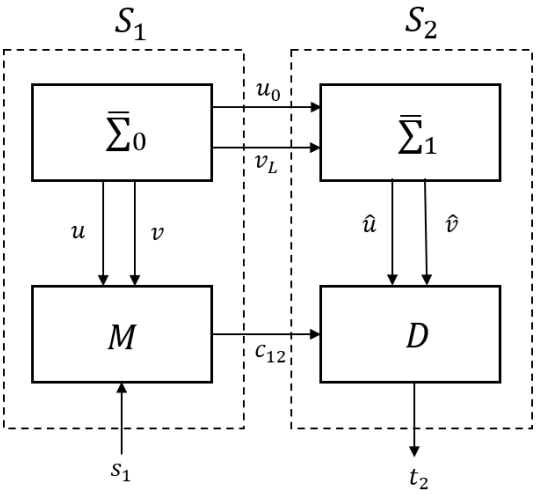

Figure 8. Secure communication system.

Modulation

$$M : \begin{cases} w(k+1) = G(u(k), v(k), w(k), s_1(k+1)) \\ c_{12} = w(k), \end{cases} \tag{24}$$

Demodulation

$$D : t_2(k+1) = G^{-1}(\hat{u}(k-1), \hat{v}(k-1), c_{12}(k-1), c_{12}(k)), \tag{25}$$

where G is the map $G : \mathbb{R}^{L+1} \times \mathbb{R}^{L+1} \times \mathbb{R}^{L+1} \times \mathbb{R}^{L+1} \to \mathbb{R}^{L+1}$ and, for an arbitrarily fixed $a, b, c \in \mathbb{R}^{L+1}$, $G(a, b, c, \cdot)$ is assumed to have an inverse map $G^{-1}(a, b, c, \cdot)$. In addition, the following conditions are also assumed to be satisfied:
(Condition 1:) For any positive number ε, there exists δ such that if

$$\|\xi_1 - \xi_2\|_{\mathbb{R}^{L+1}} < \delta \qquad (\xi_1, \xi_2 \in \mathbb{R}^{L+1}),$$

then the following inequality holds:

$$\sup_{a,b,c \in \mathbb{R}^{L+1}} \|G^{-1}(a, b, c, \xi_1) - G^{-1}(a, b, c, \xi_2)\|_{\mathbb{R}^{L+1}} < \varepsilon.$$

Using the synchronization system $\sum_1$, after the $k = L$ step, $\hat{u}$ and $\hat{v}$ are synchronized to u and v, respectively, so that after the same time, under Condition 1, the restored signal $t_2(k+1)$ is synchronized to the transmitted signal $s_1(k)$. If the original image is encrypted to $(M+1) \times (L+1)$ pixel image data and is sent from subsystem S_1 to subsystem S_2. After the running time needed for the synchronization has passed, the original image data are sliced to $s_1(k) \in \mathcal{R}^{L+1}$ (one row at a time), hence the transmission operation should be performed M times. At the reception side, the image of $(M+1) \times (L+1)$ pixels is obtained by storing the restored data $t_2(k) \in \mathcal{R}^{L+1}$ in sequence.

Remark 1. *Although the chaotic neural network in the above has the seven layers and is constructed by learning the van der Pol equation, in the proposed method, any neural network can be used as long as it is chaotic. The information required for decryption is the parameters of the neural network representing the boundary conditions. More specifically, the parameters are the matrices and the*

bias vectors that represent the linear transformations performed in each layer. If the numbers of input and output variables in a certain layer are n_{in} and n_{out}, then the number of parameters in this layer is $n_{\text{in}} \times n_{\text{out}} + n_{\text{out}}$. The sum of this number for each layer is the total number of the parameters, which represents the size of the key space. Since the number of the layers and that of the parameters can be changed freely, the size of the key space can be arbitrarily large. However, the larger the neural network becomes, the more difficult it is to be trained. Therefore, there is a trade-off between the security performance and the computational complexity of training.

Remark 2. *The computational cost of the proposed method is as follows. First, the neural network needs to be trained in advance. The time required for this is difficult to estimate because it depends on the quality of the actual data; however, for example, in the following numerical experiments, the average computation time was 1 min and 48 s for the five trials when using Google Colaboratory and Tesla P100. In addition, this pre-training has to be performed once before encryption and decryption. The time required for encryption is the same as the time required to compute a solution of the wave equation. The solution of the wave equation at each position and time can be obtained in a constant time. Hence, the computational cost is proportional to the image size. However, in reality, the computation is much more efficient than this estimation because the computation of the solution to the wave equation can be parallelized in the spatial direction, and the number of processors, especially in GPUs, typically exceeds the numbers of vertical and horizontal pixels in the image. Therefore, actually, encryption can be expected to be possible within a computation time that is several times shorter for the vertical and horizontal sizes of the image. The same is true for decryption.*

2.4. Numerical Experiments

To evaluate the proposed method, numerical experiments were conducted using the neural networks, the training data and the parameters with the following structure.

For the training data in the experiment, the input x was chosen at equal intervals from the interval $(-2, 2)$, and Equation (15),

$$\beta(u - v)^3 + (1 - \alpha)(u - v) + 2v = 0,$$

was solved using the Newton method, with the target y of the neural network being the solution to the equation. The number of data N_d was set to 20,000. The scatter plot of x and the solutions y for the equation are shown in Figure 9. To reduce the bias of the input data $N = \{x_1, x_2, ..., x_{N_d}\}$ for training, we split the dataset randomly using the train test split function of the Python library Scikit-learn to select the training data and the test data. Here, we use 20% of all data as a test set, so the number of data in the training set is 16,000 and the number of test sets is 4000. The neural network was implemented using PyTorch, and it was executed on a Tesla K80 on Google Colaboratory.

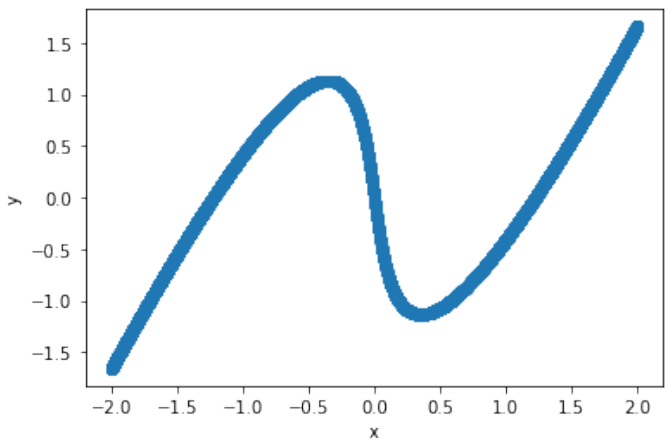

Figure 9. Scatter plot of the relationship between x and y.

The neural network used in this study was a seven layer perceptron with batch normalization layers (the reasons why the batch normalization layers are used are explained later with data in Tables 1 and 2).

Each layer has a weight W and a bias b and performs a nonlinear calculation as follows:

$$\hat{y} = g(Wx + b). \tag{26}$$

As the training depth of the neural network increases, we apply a batch normalization (BN) to each layer of the neural network as shown in Figure 10. This is to keep the overall distribution of each layer from biasing towards the upper and lower limits of the interval of the nonlinear function, which leads to the disappearance of the gradient. BN forces the distribution of the input values to follow a standard normal distribution, making the overall learning process more stable. Therefore, each mini-batch for learning should be normalized. For each mini-batch $\mathbb{B} = \{x_1, x_2, \ldots, x_m\}$, consisting of m data, the average $\mu_{\mathbb{B}}$ and variance $\sigma^2_{\mathbb{B}}$ of the mini-batch are:

$$\mu_{\mathbb{B}} = \frac{1}{N_d} \sum_{i=1}^{N_d} x_i, \tag{27}$$

$$\sigma^2_{\mathbb{B}} = \frac{1}{N_d} \sum_{i=1}^{N_d} (x_i - \mu_{\mathbb{B}})^2. \tag{28}$$

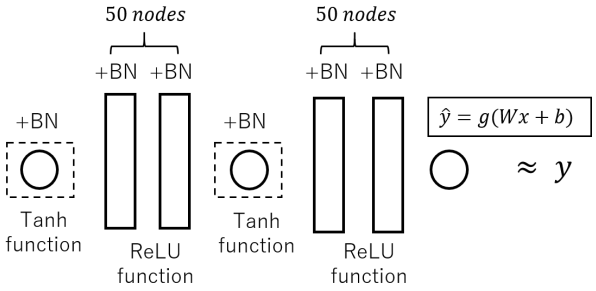

Figure 10. Structure of neural networks in numerical experiments.

Batch normalization transforms each data x_i in the mini-batch as follows:

$$\hat{x}_i = \frac{x_i - \mu_{\mathbb{B}}}{\sqrt{\sigma^2_{\mathbb{B}} + \epsilon}}, \tag{29}$$

$$y_i = \gamma \hat{x}_i + \beta, \tag{30}$$

where γ and β are the parameters of the model so that the output of each layer is normalized. In nonlinear calculations (24), $g(\cdot)$ is the activation function. Since we assumed that the images are grayscale, the first and fourth layers are set as neural network layers with only one node. In these layers, the hyperbolic function (tanh),

$$g(r) = \tanh(r) = \frac{e^r - e^{-r}}{e^r + e^{-r}}, \tag{31}$$

is the activation function so that the output converts the input value to a number in the range of -1.0 to 1.0. For layers 2, 3, 5 and 6, 50 nodes were set up and computed using the rectified linear unit (ReLU)

$$g(r) = \begin{cases} 0, & for \quad r < 0 \\ r, & for \quad r \geq 0. \end{cases} \tag{32}$$

Training error, which indicates learning accuracy, and test error, which determines generalization performance, were evaluated using the mean square error (MSE),

$$J = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2, \tag{33}$$

where $\hat{y}_i$ is the output value and y_i is the target value. We used Adam as the optimization algorithm for parameter updates in training [28]. The learning rate was set to 0.001 and the number of learning epochs was set to 1000. The training and testing errors, for example, are shown in Figures 11 and 12. Since the training results depend on the random numbers used for parameter initialization, we ran the training ten times while changing the seed of the random numbers, and evaluated the performance of the neural network using the mean and standard deviation of the results. Here, we performed two sets of experiments, one in which each layer of the neural network was nonlinearly transformed using the activation function only, and the other in which batch normalization (refer to BN) layers were added to each layer while performing the nonlinear transformation. The performance results of these two groups are shown in Tables 1 and 2, respectively.

Table 1. Mean ± standard deviation of results performed 10 times without BN.

Epoch	Data Set	Mean ± Standard
1000 times	train set	0.000005 ± 0.000001
	test set	0.000005 ± 0.000001
Lyapunov exponent		−0.067282 ± 0.102745

Table 2. Mean ± standard deviation of results performed 10 times with BN.

Epoch	Data Set	Mean ± Standard
1000 times	train set	0.000002 ± 0.000001
	test set	0.000275 ± 0.000394
Lyapunov exponent		0.024377 ± 0.148675

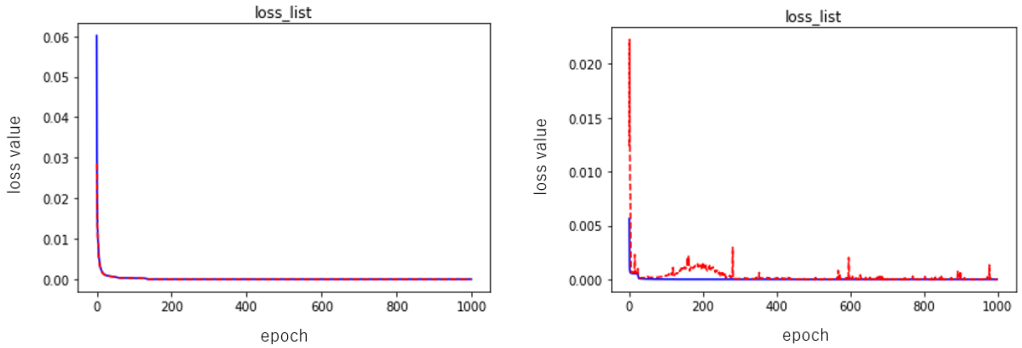

Figure 11. Examples of training error (blue line) and testing error (red line) of the neural network during the learning process. The figure on the left is the result of error training without using BN; the figure on the right is the result of error training using BN.

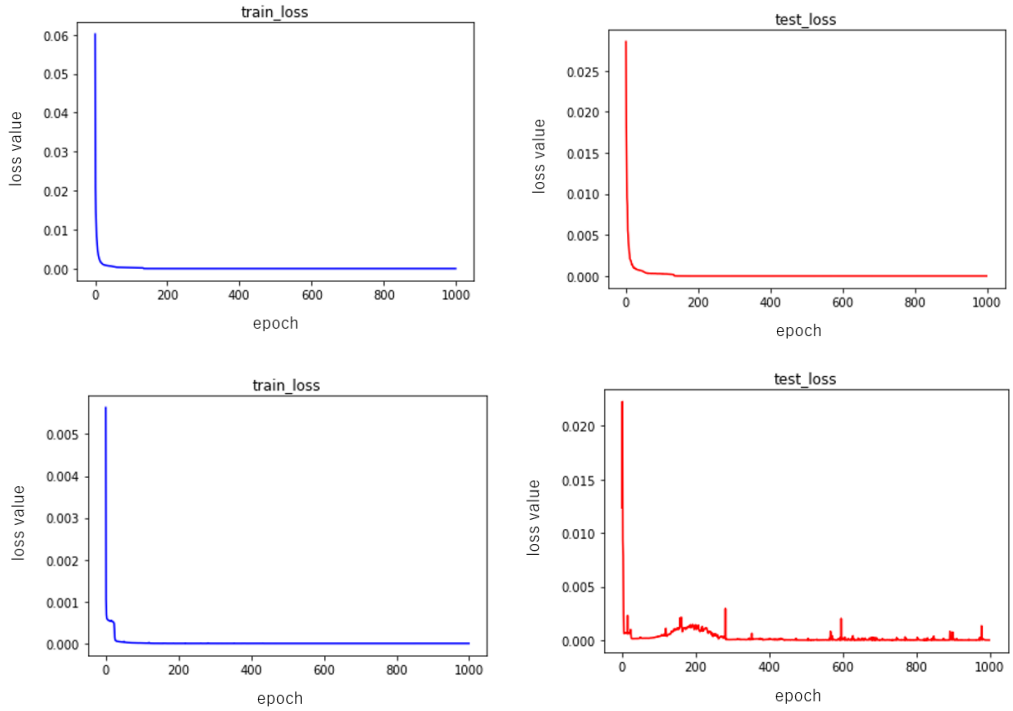

Figure 12. Examples of training error drop and test error drop.The 1st figure is training error without BN; the 2nd figure is testing error without BN; the 3rd figure is training error with BN; the 4th figure is testing error with BN.

As can be seen from Figure 11, neither the addition of the batch normalization layers nor the absence of the batch normalization layer affects the speed of the descent, and from the given illustration alone, it can also be said that the descent of the experimental group without the batch normalization layers is slightly better. Then, in terms of the values used for evaluation, the training and testing errors of the experimental group without batch normalization layers were 0.000005 ± 0.000001 and 0.000005 ± 0.000001, while the group with batch normalization layers had an error drop of 0.000002 ± 0.000001 and 0.000275 ± 0.000394, so we can say that the neural network with batch normalization layers was slightly better than the one without batch normalization layers. This is one reason why we used batch normalization.

Another important reason is the Lyapunov exponent. Introduced at the beginning of this section, the Lyapunov exponent is a numerical value used to judge whether a system is chaotic. Therefore, we also computed the corresponding Lyapunov exponent for the ten experiments, also taking the mean and standard deviation values for the numerical evaluation. From the results presented in Table 2, the average value of the Lyapunov exponent for the neural network without batch normalization layers is -0.067282, which is less than 0, indicating that this set of trained neural nets does not approximate the chaotic vibrations well. However, the average value of the Lyapunov exponent for the neural network containing the batch normalization layers is 0.024377, which is greater than 0. From this point of view, it can be said that the neural network successfully approximates the boundary conditions, thereby equipping the chaotic behaviors, and hence being suitable for the secret communication system.

We applied the proposed secret communication system to the image shown in Figure 13 with size 512×512 pixels, i.e., $M = 512, L = 512$.

Modulation

$$\begin{aligned} w(k+1) &= C(w(k))\{0.03m|u(k)|_V + 0.03m|v(k)|_V\} + \{0.5C(w(k)) + 0.1I\} \\ &\quad \times \{0.08m|u(k)|_V + 0.08m|v(k)|_V + s_1(k+1)\}, \\ c_{12}(k) &= w(k). \end{aligned} \tag{34}$$

Figure 13. Original grayscale image.

The detailed formulas for modulation and demodulation are shown below, respectively.

Demodulation

$$\begin{aligned} t_2(k+1) = \frac{1}{m}\{0.5C(c_{12}(k-1)) + 0.1I\}^{-1} \\ \times \{c_{12}(k) - C(c_{12}(k-1)) \times \{0.03m|\hat{u}(k-1)|_V + 0.03m|\hat{v}(k-1)|_V\}] \\ - 0.08|\hat{u}(k-1)|_V - 0.08|\hat{v}(k-1)|_V, \end{aligned} \tag{35}$$

where $w(k) \in \mathbb{R}^{L+1}$, $s_1(k) \in \mathbb{R}^{L+1}$, I is the $(L+1)$ order unit matrix, and $m \in \mathbb{R}$ is the parameter. We also denote $C(f) = \text{diag}(|f_0(1-f_0)|, \ldots, |f_L(1-f_L)|)$ for a vector $f = [f_0, f_1, \ldots, f_L]^T$, and $|f|_V = [|f_0|, |f_1|, \ldots, |f_L|]^T$. s_1 represents the information to be sent. $c_{12} = w$ is the information sent and received between the systems during communication, and t_2 is the information retrieved by demodulation and corresponds to s_1 if the two systems are synchronized.

Figure 14 shows the results of numerical experiments using a system with boundary conditions set by the learned neural network, with the parameters $m = 6$ in the modulation and demodulation sections. The transmitted image is shown in Figure 13. Figure 14 (left) shows the modulated image after passing through the secret communication system, and Figure 14 (right) shows the image recovered by the synchronous system. In addition, $m = 6$ is a parameter value obtained after numerous attempts, and the size of m affects the outcome of the entire modulation and demodulation operations (see Figure 15).

Figure 14. Encrypted grayscale images and restored images in the proposed method of this study. The figure on the left is the encrypted grayscale image; the figure on the right is the restored image.

As shown in Figure 13, the original image is almost unrecognizable from the encrypted image. Figure 14 (right) shows the reconstructed image after the demodulation section. There is little distinction between the transmitted image and the reconstructed image.

Figure 15. The effect of changing the m value in the modulation and demodulation portions on the image encryption effect. Six different values of m (m = 0.8, 1.25, 2.5, 3.0, 4.5, 6.0) were tried, and the corresponding encryption effects were observed for each, where the larger the value of m, the better the secrecy performance of the image, within the computable range.

3. Numerical Experiments with Color Images

In the previous section, we explained the wave equations with van der Pol boundary conditions and the corresponding synchronization systems. We then improved the security of the confidential communication system by approximating the chaotic phenomena with a neural network; however, relatively simple grayscale images are used as input objects. Next, we will replace the input images with colored ones and conduct experiments to see if the proposed approach is still applicable to those images. The biggest difference between a color image and a grayscle image is that each pixel of a color image is usually represented by three components, red (R), green (G) and blue (B), of which intensities are represented by numerics between, for example, (0, 255), while a pixel of a grayscale image has a single component. This results in three differences: one being the structure of the neural network itself, the second being that the training method for each pixel is also optional, which means the three components (R, G and B) of each pixel can be trained separately by themselves or mixed together, and the third being the Lyapunov exponent for testing chaotic phenomena.

Color images in the neural network can be computed, one by one, as

$$x_r \to f(x_r) \tag{36}$$
$$x_g \to f(x_g) \tag{37}$$
$$x_b \to f(x_b). \tag{38}$$

The neural network structure in this approach is still one input neuron and one output neuron; they are represented by (36)–(38) and are trained separately. However, the security of this approach is not very high and there remains a risk of theft. Therefore, in this study, after R, G and B are fed into the neural network, we will mix and disrupt them, adopting a structure that has three input neurons and three output neurons, thus improving the security of the confidential communication system.

The composition diagram of applying RGB to the neural network is shown in Figure 16, where the data from three neurons are used as inputs to the input layer and then sent to the hidden layer. First, since color images are involved, $\sum_0$ and $\sum_1$ correspond to each color in RGB, and the dependent variables u and v are expanded to three dimensions. The u and v are then discretized according to the upwind difference method to obtain the values of u, v at the next time step. (For this part of the process we can refer to Figure 5 in Section 2.3). Because the number of inputs is increased from one to three, the number of the nodes of the first, middle and the last layers are also increased accordingly. By using the input image, the x_r, x_g, and x_b are computed by using chaotic maps as follows:

$$\begin{matrix} x_r \searrow & & \\ x_g \to g_i(x_r, x_g, x_b) \to f(g_i(x_r, x_g, x_b)) & (i = 1, 2, \ldots, h_n), \\ x_b \nearrow & & \end{matrix} \tag{39}$$

where h_n is the number of neurons in the hidden layer. The former half of the neural network, F_0, is treated as the left boundary condition, and the latter half, F_1, is treated as the right boundary condition:

$$\begin{aligned} u(1,t) &= F_1(v(1,t)) \\ v(0,t) &= F_0(u(0,t)). \end{aligned} \tag{40}$$

We know that the existence of chaos can be determined by calculating the Lyapunov exponent of the dynamical system. If the images are grayscale, the Lyapunov exponent has just one value, hence whether the system is chaotic or not can be determined by examining the exponent. When confronted with color images, the Lyapunov exponent is no longer

1-dimensional due to dimensional growth. k-dimensional space has k Lyapunov exponents. In fact, the Lyapunov exponent $\{\lambda_1, \lambda_2, \ldots, \lambda_k\}$ is defined as

$$(e, e^{\lambda_1}, e^{\lambda_2}, \ldots, e^{\lambda_k}) = \lim_{N\to\infty} [\text{magnitude of the eigenvalues of } \prod_{n=0}^{N-1} J(x_n)^{1/N}], \tag{41}$$

and

$$J(x_n) = (\frac{\partial G_i}{\partial x_i}) \tag{42}$$

is the Jacobian matrix of the map $G(x_n)$. Normally, if the maximum Lyapunov exponent λ_1 is positive, the system can be regarded as chaos, but even if $\lambda_1 < 0$, there may be a latent chaotic case which cannot be observed [27]. In this study, as long as one of the Lyapunov exponents is positive, we conclude that the neural network contains chaotic vibrations, which means that the approximation of the chaotic boundary condition is successful.

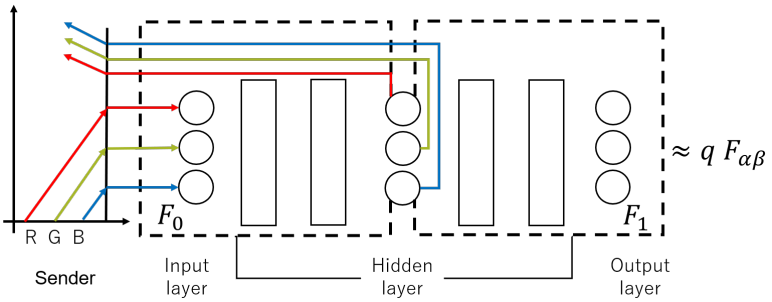

Figure 16. Relationship between the structure of Neural Networks and boundary conditions with RGB.

For the numerical experiments, we first trained the neural networks so that the chaotic boundary condition is approximated. We choose training data, from which input data are sampled from $(-2, 2)$. Since a pixel has three components, the total amount of data should preferably be a multiple of 3; we choose $N_d = 30{,}000$ and (15) is again solved by the Newton method. The data are randomly divided into a training set and a test set in the ratio of 8:2. The neural network was implemented using PyTorch, and it was executed on a Tesla T4 from Google Colaboratory.

The structure of the neural network is still a seven layer multilayer perceptron with three neurons in the first, fourth and seventh layers, and 50 neurons in the other hidden layers. Each layer is nonlinearly transformed by $\hat{y} = g(Wx + b)$ and accompanied by BN, where the activation functions of the first and fourth layers are tanh functions and the others are ReLU functions. The mean squared error is still used for the error function. The entire training process was updated with the Adam algorithm with a learning rate of 0.001. The training was completed after 1000 training runs.

Table 3 shows the numerical evaluation of the training results and the test results of the neural network. A total of ten experiments were conducted, and each time the seed value was changed so that the initial values of the parameters of the neural networks are changed. The mean and standard deviation are computed for the experimental results. We can see that the results of training and testing are 0.000694 ± 0.000035 and 0.000919 ± 0.000033, respectively, and Figure 17 shows the error decrease of one of the trainings, from which we can clearly see that the training error and the testing error both decrease rapidly, and there is almost no error rebound. Then, we calculated the Lyapunov exponent, respectively, $\lambda_1 = 0.1144 \pm 0.1469$, $\lambda_2 = 0.0038 \pm 0.0917$ and $\lambda_3 = -0.1717 \pm 0.2333$, where λ_1 and λ_2 are positive, so we can conclude that the neural networks are in fact chaotic, which somehow approximates the van der Pol boundary condition.

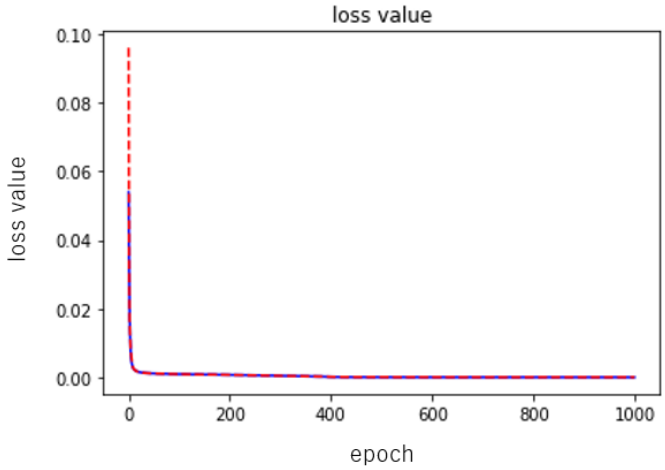

Figure 17. Examples of training error (blue line) and testing error (red line) of the neural network during the learning process.

Table 3. Mean ± standard deviation of results performed at 10 times with RGB.

Epoch	Data Set	Mean ± Standard
1000 times	train set	0.000694 ± 0.000035
	test set	0.000919 ± 0.000033
Lyapunov exponent		0.1144 ± 0.1469
		0.0038 ± 0.0917
		−0.1717 ± 0.2333

A color image of size 512 × 512, as shown in Figure 18, is put into the whole system as the input. The modulation and demodulation are shown below.

Figure 18. Original color image.

Modulation

$$\begin{aligned} w(k+1) &= C(w(k))\{0.03m|u(k)|_V + 0.03m|v(k)|_V\} + \{0.5C(w(k)) + 0.1I\} \\ &\quad \times \{0.08m|u(k)|_V + 0.08m|v(k)|_V + s_1(k+1)\}, \\ c_{12}(k) &= w(k). \end{aligned} \tag{43}$$

Demodulation

$$t_2(k+1) = \frac{1}{m}\{0.5C(c_{12}(k-1)) + 0.1I\}^{-1} \times \{c_{12}(k) - C(c_{12}(k-1)) \times \{0.03m|\hat{u}(k-1)|_V + 0.03m|\hat{v}(k-1)|_V\}] - 0.08|\hat{u}(k-1)|_V - 0.08|\hat{v}(k-1)|_V. \quad (44)$$

With (43) and (44), we can modulate and recover the images; however, the parameter m still needs to be tried. In Figure 19, we have tried six different values of m. It can be seen that the greater the value of m, the better the secrecy of the image. Figure 18 shows the experimental results when the value of $m = 8.8$. Figure 20 (left) is the image that has been secreted by the proposed system, from which we can conclude that the proposed method is successfully applied to the transmission of color images. In fact, it is very difficult to know the original image from the transmitted one while the restored image shown in Figure 20 (right) is almost identical to the original color image.

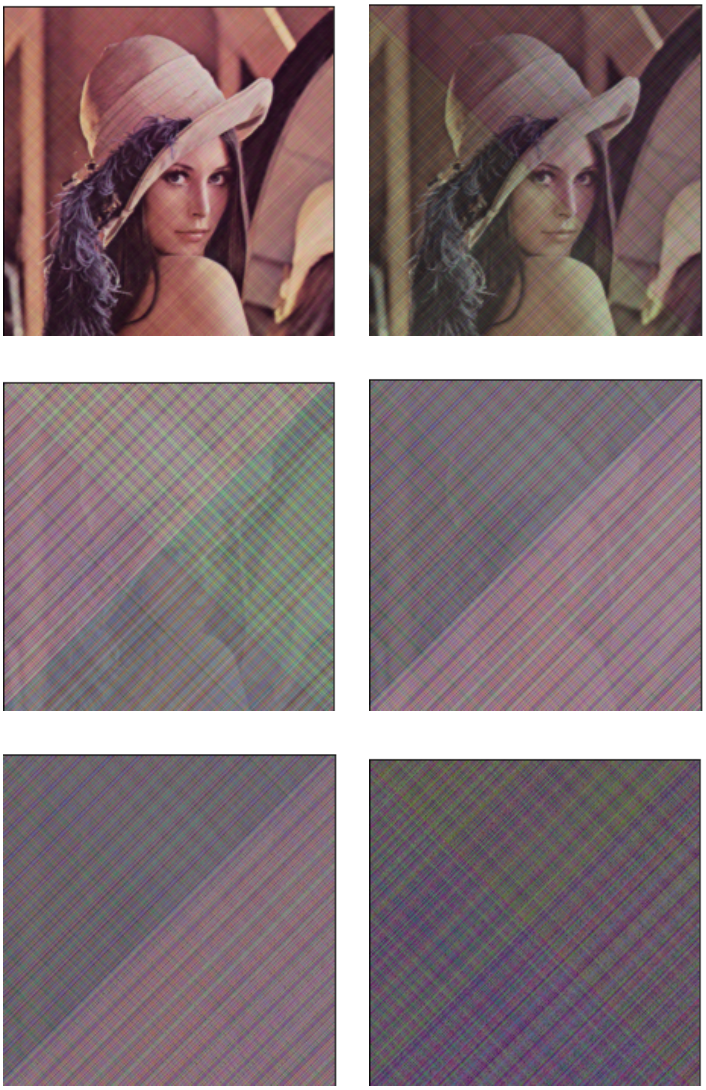

Figure 19. The effect of changing the m value in the modulation and demodulation portions on the image encryption effect. Six different values of m (m = 1.0, 2.0, 5.0, 6.0, 7.5, 8.8) were tried, and the corresponding encryption effects were observed for each, where the larger the value of m, the better the secrecy performance of the image, within the computable range.

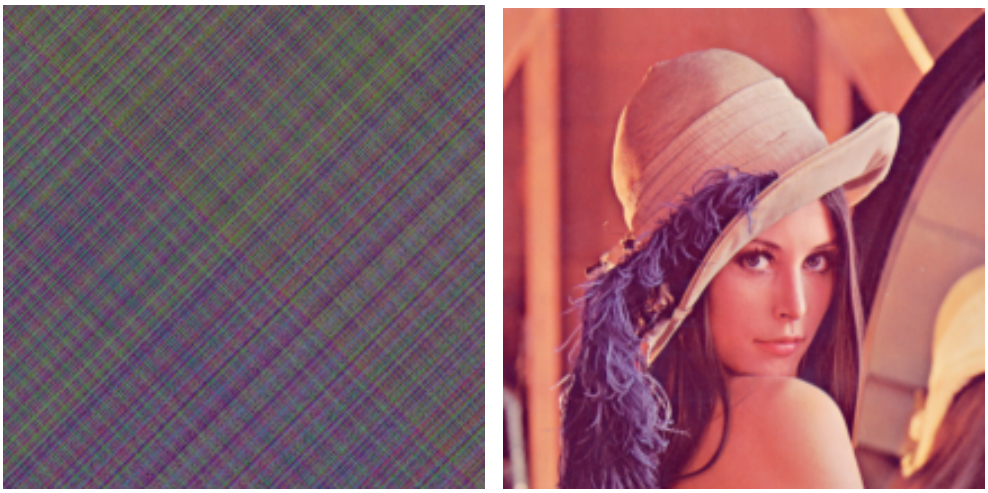

Figure 20. Encrypted color image and restored image in the proposed method of this study. The figure on the left is the encrypted color image; the figure on the right is the restored color image.

4. Security Evaluation of the Encrypted Images by the Proposed Method

In the previous section, we numerically examined the proposed method in which the chaotic synchronization system and the artificial neural network are combined. In particular, we tested different values of m to observe the efficiency of the encryption and decryption of the images. In this section, we test the security of the encrypted images. For some different images and the values of m, we investigate whether the encrypted images are secure or not by computing several numerical security measures and color histograms.

Based on traditional evaluation ideas, we first test the randomness of encrypted images: the correlation coefficient and the UACI value [29]. We compare the proposed method with Advanced Encryption Standard (AES) as a reference method. AES is a traditional encryption method, which is a group cipher, where the plaintext is divided into groups of equal length and each group is encrypted until the entire plaintext is encrypted. We used AES in the ECB mode.

Although these are standard measures of security, note that the above measures are not suitable for the proposed approach. In fact, in order to choose suitable measures, the term of "ideal encrypted image" needs to be specified [30]. In typical statistical measures, encrypted images are considered to be secure for, for example, differential attacks when it exhibits randomness. On the other hand, the proposed approach is not designed to generate pseudo-random sequences because a different criterion is used for the term "ideal encrypted image". The proposed approach aims to encrypt the images into almost identical wave images, which are considered to be "ideal" in this study. The idea behind this is that because, for example, the differential attacks essentially try to find how differences in the original images affect the encrypted images, if the encrypted images are almost identical, then the attacks should fail and the information on the original image cannot be retrieved from them.

Although the statistical measures are not necessarily suitable for the proposed method, to a certain extent, the method has a good statistical property as shown below. In addition, we also performed more appropriate tests, in which the distinguishability of two encrypted images is checked.

4.1. Randomness Testing of Encrypted Images

Firstly, we computed the correlation coefficients between adjacent pixels for the encrypted image in three different directions: horizontal, vertical and diagonal. Suppose

that the image has $N \times N$ pixels (x_i, y_j), $i = 1, \ldots, N$, $j = 1, \ldots, N$. We calculate the correlation coefficients for R, G and B values, respectively:

$$r_{xy} = \frac{\frac{1}{N}\sum_{i=1}^{N}\sum_{j}^{N}(x_i - E(x)) \times (y_i - E(y))}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2}\sqrt{\frac{1}{N}\sum_{j=1}^{N}(y_j - E(y))^2}}, \tag{45}$$

where $E(x) = \frac{1}{N}\sum_{i}^{N} x_i, E(y) = \frac{1}{N}\sum_{j}^{N} y_j$. The correlation coefficients are between -1 and 1. For example, when the value is close to 1, it means a high correlation between pixels. When it is close to 0, there is no correlation; hence the image is considered pseudo-random. The results are shown in Table 4 for the grayscale images and in Table 5 for the color images. The images encrypted by the proposed method all present a high inter-pixel correlation, while the traditional AES encryption has an ideal value of around 0. This is because the encrypted images of the proposed method have a certain regularity that is caused by the fact that the images represent the trajectories of waves. Note that this regularity does not imply the recognizability of the original images.

Secondly, we computed UACI values, which measure how much the encrypted image differs from the original image. For two different images, I_1 and I_2, of the same size $N \times N$, the UACI values between them is defined by:

$$UACI = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{I_1(i,j) - I_2(i,j)}{\text{tonal range}} \times 100\%. \tag{46}$$

Considering the color range and gray value distribution of images, the ideal value is found to be 33.3. The closer the UACI of the encrypted image is to this ideal value, the better the security.

The computed values are shown in Table 4 for the grayscale images and in Table 6 for the color images. All the results for AES are about 50.0, while those for the proposed method depend on the target images. AES performs better for the images of vegetables; however, the values of the proposed method for the images of the woman are better than those of AES.

Thirdly, we observed the histograms, which are graphical representations of the intensity distribution of pixels in an image. For gray images, the grayscale histogram reflects the grayscale statistical information of the image. For example, each grayscale image of the previous numerical tests has 256 intensity levels with values from 0 to 255. We store the number of pixels corresponding to each gray level in a 256 capacity array. Similarly, for color images, we can compute the histograms for each of the three different channels, R, G and B, respectively.

The results are shown in Figure 21 for the grayscale images and in Figure 22 for the color images. We can see that the histograms of the images encrypted by AES show a uniform distribution, both in color and grayscale, which to some extent indicates that the encrypted images are disordered and hence in a secure state.

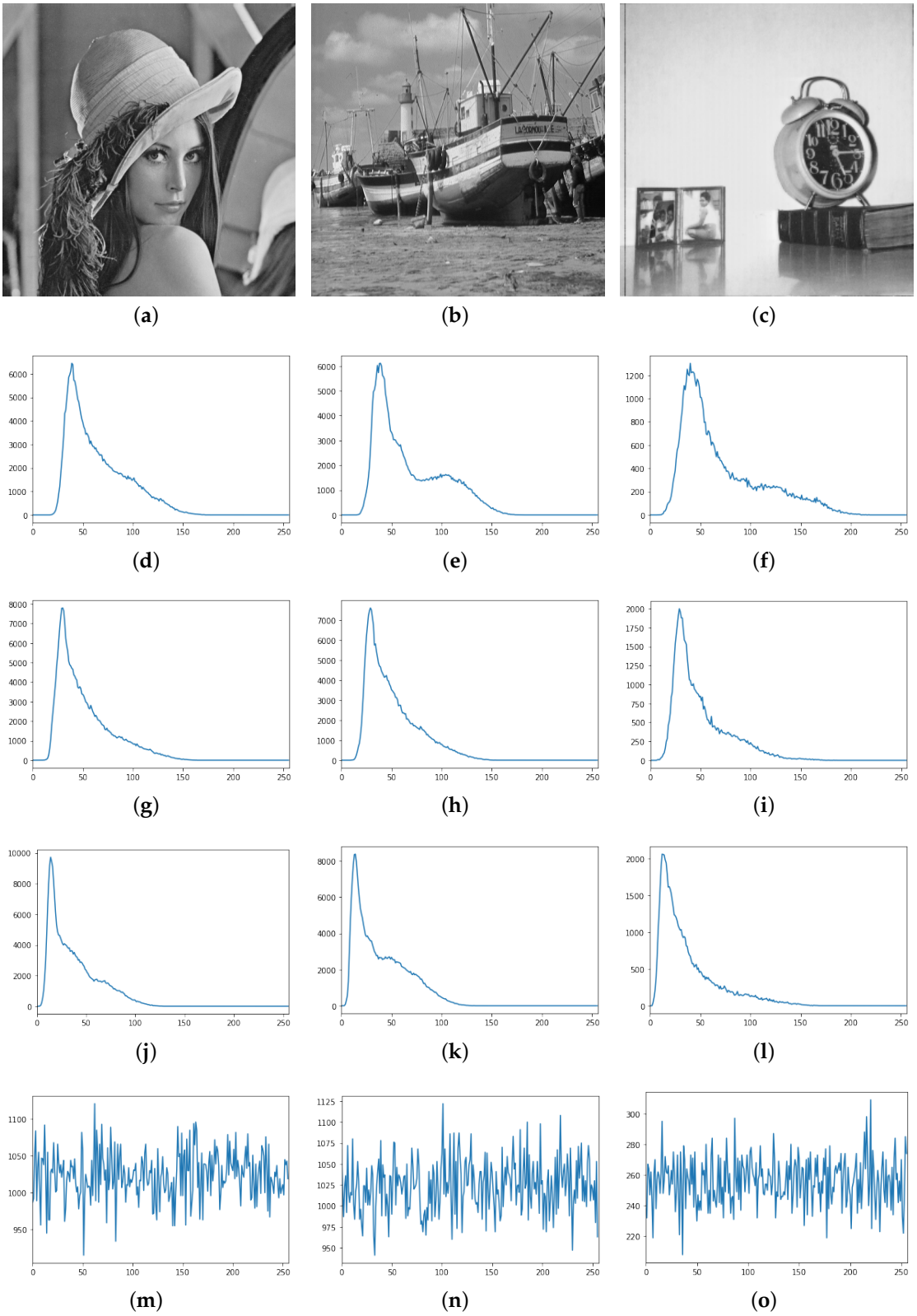

Figure 21. The grayscale images and the histograms of the encrypted images. The first row shows three different grayscale images (**a**–**c**). In the second, third and fourth rows, the three figures in each row represent the histograms of the grayscale values of the three original images encrypted by the proposed approach under the settings of $T = 1, 2, 3$ and $m = 6.0$, respectively (**d**–**l**). The last row shows the histogram after encryption by the AES. The horizontal coordinate represents the tonal range and the vertical coordinate represents the absolute frequency (**m**–**o**).

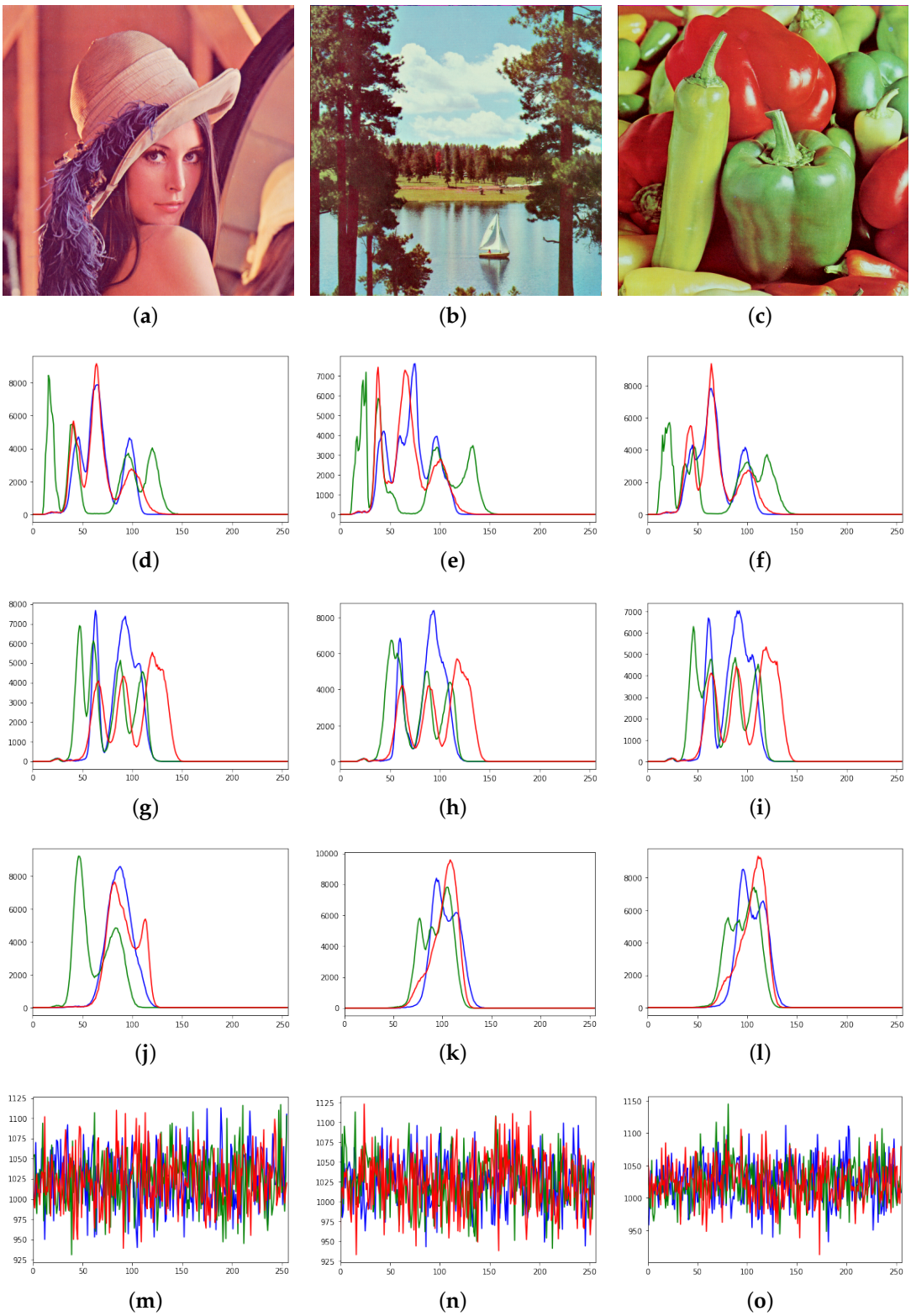

Figure 22. The color images and the histogram of the encrypted image. The first row shows three different color images (**a**–**c**). In the second, third and fourth rows, the three figures in each row represent the histograms of the three original images encrypted by the proposed approach under the settings of $T = 1, 2, 3$ and $m = 6.0$, respectively. The last row shows the histogram after encryption by the AES. The red, green and blue colors correspond to the histograms of each channel of R, G and B, respectively (**d**–**l**). The last row shows the histograms after encryption by the AES. The horizontal coordinates represent the tonal range and the vertical coordinates represent the absolute frequency (**m**–**o**).

Table 4. Correlation test in horizontal, vertical and diagonal directions and UACI test for different grayscale images. Different m ($m = 6, 7.5$) and T ($T = 1, 2, 3$) were tried respectively, where "lena" is Figure 21a, "boat" is Figure 21b, and "clock" is Figure 21c.

Object			Correlation			UACI
			Horizontal	Vertical	Diagonal	
lena	$m = 6$	$T = 1$	0.972642	0.972993	0.922492	36.141486
		$T = 2$	0.962627	0.962568	0.890740	39.410799
		$T = 3$	0.968880	0.969089	0.915038	35.114740
	$m = 7.5$	$T = 1$	0.968823	0.969244	0.908995	39.877836
		$T = 2$	0.954396	0.954471	0.874803	39.730604
		$T = 3$	0.976311	0.976517	0.940879	35.622222
	AES		0.002630	0.008785	0.000658	49.996347
boat	$m = 6$	$T = 1$	0.980538	0.980823	0.943327	36.728787
		$T = 2$	0.961264	0.961193	0.885866	41.783973
		$T = 3$	0.973791	0.973994	0.930897	39.928194
	$m = 7.5$	$T = 1$	0.972897	0.973300	0.923706	36.761638
		$T = 2$	0.957479	0.957473	0.874213	41.551513
		$T = 3$	0.973904	0.974118	0.936444	40.359627
	AES		0.000163	0.000445	0.000502	50.000055
clock	$m = 6$	$T = 1$	0.921742	0.922687	0.806450	48.089881
		$T = 2$	0.858309	0.858372	0.664064	58.067992
		$T = 3$	0.888560	0.889011	0.722784	58.823428
	$m = 7.5$	$T = 1$	0.899274	0.900966	0.770316	47.179087
		$T = 2$	0.853312	0.957473	0.874213	56.637741
		$T = 3$	0.880133	0.881703	0.754349	57.388910
	AES		0.005367	0.004363	0.003360	49.929277

Table 5. Correlation test in horizontal, vertical and diagonal directions with different color images. Different values of m ($m = 7.5, 8.8$) and T ($T = 1, 2, 3$) were tried respectively, where "lena" is Figure 22a, "boat" is Figure 22b, and "veg" is Figure 22c.

Object			Correlation								
			Horizontal			Vertical			Diagonal		
			R	G	B	R	G	B	R	G	B
lena	$m = 7.5$	$T = 1$	0.97	0.99	0.98	0.97	0.99	0.98	0.95	0.99	0.96
		$T = 2$	0.99	0.99	0.98	0.99	0.99	0.98	0.98	0.98	0.97
		$T = 3$	0.93	0.94	0.81	0.93	0.94	0.81	0.82	0.89	0.59
	$m = 8.8$	$T = 1$	0.96	0.99	0.97	0.96	0.99	0.97	0.88	0.88	0.88
		$T = 2$	0.98	0.99	0.98	0.98	0.99	0.97	0.96	0.98	0.96
		$T = 3$	0.89	0.95	0.82	0.90	0.95	0.83	0.75	0.88	0.57
	AES		5×10^{-3}	1×10^{-3}	-1×10^{-3}	1×10^{-2}	6×10^{-3}	8×10^{-3}	1×10^{-3}	-8×10^{-5}	3×10^{-3}

Table 5. *Cont.*

Object			Correlation								
			Horizontal			Vertical			Diagonal		
			R	**G**	**B**	**R**	**G**	**B**	**R**	**G**	**B**
boat	$m = 7.5$	$T = 1$	0.96	1.00	0.98	0.97	1.00	0.98	0.94	0.99	0.96
		$T = 2$	0.99	0.99	0.98	0.99	0.99	0.98	0.98	0.98	0.98
		$T = 3$	0.92	0.97	0.92	0.93	0.97	0.92	0.82	0.94	0.86
	$m = 8.8$	$T = 1$	0.96	0.99	0.97	0.96	0.99	0.97	0.93	0.99	0.95
		$T = 2$	0.98	0.99	0.97	0.98	0.99	0.97	0.96	0.98	0.96
		$T = 3$	0.92	0.96	0.92	0.93	0.97	0.93	0.82	0.93	0.87
	AES		6×10^{-4}	-1×10^{-3}	-1×10^{-4}	-3×10^{-3}	-1×10^{-3}	-3×10^{-3}	1×10^{-3}	-2×10^{-3}	-3×10^{-3}
veg	$m = 7.5$	$T = 1$	0.97	0.99	0.98	0.97	0.99	0.98	0.94	0.99	0.96
		$T = 2$	0.99	0.99	0.98	0.99	0.99	0.88	0.98	0.98	0.97
		$T = 3$	0.92	0.97	0.92	0.93	0.97	0.92	0.82	0.94	0.86
	$m = 8.8$	$T = 1$	0.96	0.99	0.97	0.97	0.99	0.98	0.94	0.99	0.96
		$T = 2$	0.98	0.99	0.97	0.98	0.99	0.97	0.96	0.98	0.96
		$T = 3$	0.92	0.96	0.92	0.93	0.96	0.92	0.82	0.96	0.86
	AES		2×10^{-4}	2×10^{-3}	5×10^{-3}	-8×10^{-4}	2×10^{-3}	6×10^{-4}	4×10^{-4}	4×10^{-3}	7×10^{-4}

Table 6. UACI test for different color images. Different values of m ($m = 7.5, 8.8$) and T ($T = 1, 2, 3$) were tried respectively, where "lena" is Figure 22a, "boat" is Figure 22b, and "veg" is Figure 22c.

Object			UACI		
			R	**G**	**B**
lena	$m = 7.5$	$T = 1$	46.293217	40.515568	24.708355
		$T = 2$	43.678409	43.262431	37.007906
		$T = 3$	41.407206	42.583560	29.628579
	$m = 8.8$	$T = 1$	46.454520	41.186064	24.949227
		$T = 2$	44.341227	43.377668	33.920451
		$T = 3$	43.072341	40.273322	40.672362
	AES		50.133255	50.047464	49.936812
boat	$m = 7.5$	$T = 1$	35.143726	48.660561	52.421603
		$T = 2$	39.571601	55.066602	56.038358
		$T = 3$	47.716444	53.135069	53.060568
	$m = 8.8$	$T = 1$	35.245613	50.609906	55.037498
		$T = 2$	38.337034	56.456049	58.441035
		$T = 3$	42.331758	55.305545	56.647502
	AES		49.953939	50.052023	50.175874
veg	$m = 7.5$	$T = 1$	37.751291	54.587069	54.226224
		$T = 2$	45.768913	47.626056	62.656476
		$T = 3$	37.213291	48.628576	69.999379

Table 6. *Cont.*

Object		UACI		
		R	**G**	**B**
$m = 8.8$	$T = 1$	37.915899	54.778044	55.353727
	$T = 2$	62.202183	48.556424	40.446628
	$T = 3$	33.298991	51.570749	69.901463
AES		49.956022	50.026667	49.943955

Meanwhile, the histogram of the encrypted image in the mode of the proposed method presents an unbalanced state with a high concentration in a certain region. The reason for this phenomenon is as follows. The proposal method uses the reflection of waves, and even though the initial conditions are random values, they are propagated to the left and right sides along straight lines. In fact, as shown in Figure 23, the images encrypted by AES and the images encrypted by the proposed method are apparently different from each other. The former has almost no structure, and the latter a visible structure that corresponds to the trajectories of the waves. The above mentioned concentration is observed because the intensity of a trajectory takes the same value.

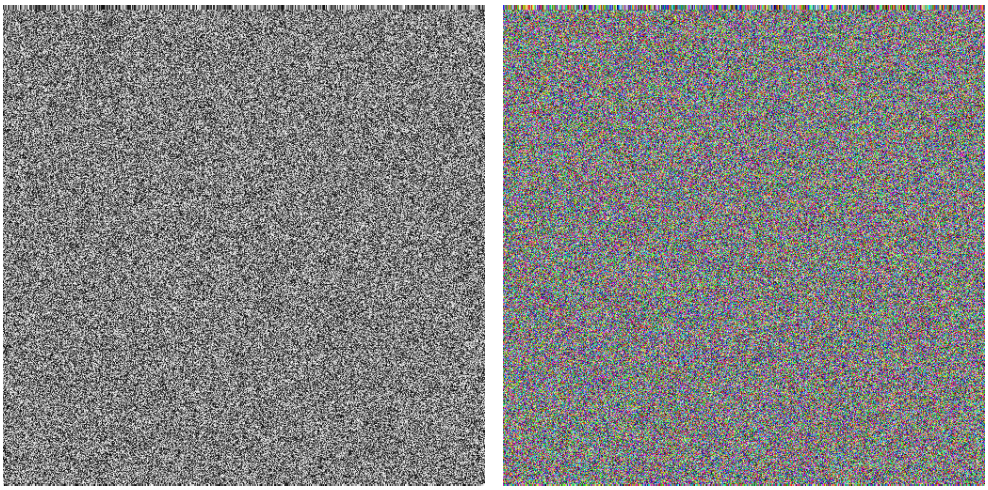

Figure 23. The grayscale encrypted image (**left**) and the color encrypted image (**right**) obtained by using the AES on the grayscale image (Figure 13) and the color image (Figure 20).

From this consideration, we deduce that the difference in the position of the highest point of the histogram reflects the difference in the lengths of the straight lines, which in turn depend on the waiting time for synchronization. To confirm this, we computed the histograms changing the waiting time as $T \times$ (the vertical size of the target image) with $T = 1, 2, 3$. For grayscale images, the highest point of the histogram increases with the waiting time, and for color images, R, G and B have multiple peaks and the position and the number of the peaks depend on the waiting time. This implies that although the histograms of the encrypted images have some peaks, from these peaks only the waiting time for synchronization can be estimated, and at least in a naive way, no information on the original images can.

4.2. Distinguishability of Encrypted Images

From the above security tests, we can say that AES has higher security from the perspective of randomness; however, the AES and the proposed approach are fundamentally different in the definition of "ideal encrypted images". Hence, it is important to investigate the security issue from another perspective.

Since the ideal encrypted images of the proposed approach are "the indistinguishable images of waves", we calculated the similarity of encrypted images of two different images. If there is a high correlation between the two, it indicates that the proposed method encrypts different images into almost identical encryption results. In addition, if two encrypted images have high similarity, there is a very low probability of leaking the original image information.

We evaluated three inter-image-similarity criteria [31,32] for color and grayscale images. Firstly we use the averaging pixel values,

$$\frac{\sqrt{(\overline{I1_R} - \overline{I2_R})^2 + (\overline{I1_G} - \overline{I2_G})^2 + (\overline{I1_B} - \overline{I2_B})^2}}{\sqrt{3 \times (255 - 0)^2}}, \tag{47}$$

where $I1, I2$ are images and the averaging value, for example, $\overline{I_R}$, for the image I is

$$\overline{I_R} = \frac{1}{N_{\text{pixel}}} \sum_{i=0}^{N_{\text{pixel}}} I_{R_i}, I_R = \{I_{R_0}, I_{R_1}, \ldots, I_{R_{N_{\text{pixel}}}}\}.$$

N_{pixel} is the number of pixels and I_{R_i} is the red value of the ith pixel of the image I. Secondly, we use the histogram for pixel values,

$$\frac{\sqrt{\sum_{i=0}^{N_{\text{int}}} (I1_{RH_i} - I2_{RH_i})^2} + \sqrt{\sum_{i=0}^{N_{\text{int}}} (I1_{GH_i} - I2_{GH_i})^2} + \sqrt{\sum_{i=0}^{N_{\text{int}}} (I1_{BH_i} - I2_{BH_i})^2}}{3 \times \sqrt{2 \times (1 - 0)^2}},$$
$$I_{RH} = \{I_{RH_0}, I_{RH_1}, \cdots, I_{RH_{N_{\text{int}}}}\}, \tag{48}$$

where $I1, I2$ are images and N_{int} is the number of the levels of intensity, and I_{RH_i} is the relative degree of the ith bin of the histogram of the red values of the image I. Thirdly, we use the correlation coefficients:

$$1 - \left| \frac{r_{I1I2_R} + r_{I1I2_G} + r_{I1I2_B}}{3} \right|, \tag{49}$$

where

$$r_{I1I2_R} = \frac{\sum_{i=0}^{N_{\text{pixel}}} (I1_{R_i} - \overline{I1_R})(I2_{R_i} - \overline{I2_R})}{\sum_{i=0}^{N_{\text{pixel}}} (I1_{R_i} - \overline{I1_R})^2 \sum_{i=0}^{N_{\text{pixel}}} (I2_{R_i} - \overline{I2_R})^2}.$$

In the experiments, the resolutions of the two images are unified into the same value. We computed these values changing the value of m and the waiting time for synchronization.

From the results of the grayscale and color images shown in Tables 7 and 8, it can be seen that the correlation coefficients between the encrypted images of different images are significantly large. The values are indeed almost close to one; when ten decimal places are used, the values of correlation coefficients are, for example, 0.9999999977 for the grayscale image of "boat" with $m = 7.5, T = 1$ and that of "lena" with $m = 6.0, T = 1$ and 0.9999999865 for the color image of "boat" with $m = 7.5, T = 1$ and that of "lena" with $m = 8.8, T = 2$. The other two measures are very small in all cases, indicating high indistinguishability between the encrypted images, hence the high security of the proposed approach.

Table 7. Similarity comparison between the grayscale encrypted images $I1$ and $I2$ with the proposed approach. Six different encrypted images are taken, the similarity is calculated between each two, and no comparison is made between encrypted images of the same original image. The result column has three values, from top to bottom, corresponding to (47)–(49). In particular, "lena" is Figure 21a, "boat" is Figure 21b, and "clock" is Figure 21c.

I1 \ **I2**	**Lena** ($m = 6$, $T = 1$)	**Boat** ($m = 7.5$, $T = 2$)	**Clock** ($m = 6$, $T = 2$)
lena ($m = 7.5$, $T = 1$)		0.068396	0.0898650
		0.112057	0.119775
		0.999999	0.999999
boat ($m = 7.5$, $T = 1$)	0.042764		0.099446
	0.077088		0.143051
	0.999999 *		0.999999
clock ($m = 6$, $T = 3$)	0.106253	0.071040	
	0.107990	0.081516	
	0.999999	0.999999	

∗: The value is 0.9999999977 when 10 decimal places are used.

Table 8. Similarity comparison between the color encrypted images $I1$ and $I2$ with the proposed approach. Six different encrypted images are taken, the similarity is calculated between each two, and no comparison is made between encrypted images of the same original image. The result column has three values, from top to bottom, corresponding to (47)–(49). In particular, "lena" is Figure 21a, "boat" is Figure 21b, and "veg" is Figure 21c.

I1 \ **I2**	**Lena** ($m = 8.8$, $T = 2$)	**Boat** ($m = 7.5$, $T = 2$)	**Veg** ($m = 8.8$, $T = 1$)
lena ($m = 8.8$, $T = 3$)		0.047194	0.070985
		0.103503	0.184824
		0.999999	0.999999
boat ($m = 7.5$, $T = 1$)	0.084080		0.008566
	0.185896		0.037862
	0.999999 *		0.999999
veg ($m = 7.5$, $T = 1$)	0.092913	0.106315	
	0.191360	0.210313	
	0.999999	0.999999	

∗: The value is 0.9999999865 when 10 decimal places are used.

5. Conclusions

The purpose of this study is to establish a more stable and secure communication system by improving the method reported in [19]. The reason for this is that the number of parameters in a confidential communication system, using the chaotic synchronization system from [19], is too small, and the image information can be easily stolen so that even if only one parameter is attacked, there is a certain chance that the original image can be seen in its entirety. In order to improve the security of the system based on the original concept, the development of a new method that includes a chaotic system with

a larger number of independent parameters is necessary. Therefore, this study proposes a confidential communication system for color image communication using a chaotic synchronous distribution system and deep learning. Deep neural networks have properties that satisfy both conditions; on the one hand, they can approximate any function, and on the other hand, they have a complicated structure so that it is difficult to steal all the information. Therefore, we use a neural network to approximate the van der Pol boundary condition, so that the learned neural network can exhibit chaotic behavior instead of the original boundary condition. In the construction of the neural network, we consider the middle layer of the hidden layer as an output layer as well, and the number of neurons in the input and output layers is related to the input image, so that the network can be divided into two parts, corresponding to the left and right boundary conditions, respectively.

We first used grayscale images as experimental objects to test whether the proposed approach is practically applicable. The neural network has one input neuron and one output neuron, and the intermediate layer is also set to one neuron, and two different activation functions, Tanh and ReLU, are used for nonlinear transformation. The experimental results are analyzed from two aspects: first, during the training and testing of the neural network itself, the error decreases very quickly and there is no error rebound, and the value of the Lyapunov exponent is also positive, which shows that the neural network can learn the van der Pol boundary conditions well to obtain the chaotic phenomenon; second, the modulated image can well mask the information of the original grayscale image, and the recovered image is the same as the original grayscale image. After confirming that the proposed approach can be applied to simpler grayscale images, we further experimented with color images. Since the pixels are expanded from one dimension to three dimensions, a three-input, three-output neural network structure is adopted for a better stealth effect. The experimental results are very close to those of the grayscale images, the neural network still learns the chaotic phenomena, and it can hide the images and restore the images well.

Several security tests were also performed. The proposed method is not designed to generate pseudo-random sequences as the encrypted images. Therefore, statistical tests were not appropriate, and in many cases AES produced images that were closer to random numbers. However, in terms of the UACI value, the proposed method was superior in some cases. On the other hand, if the encrypted images are identical regardless of the original image, then it becomes difficult to infer the original image from the encrypted image. We performed some tests from this perspective as well. The results imply that the encrypted images from the proposed method are almost the same regardless of the original images, thereby showing that the proposed method is certainly secure.

Although we have confirmed to some extent that the techniques used in this study can be applied to color images, there is still room for improvement. In particular, the neural network in this study learns chaotic phenomena by approximating $qF_{\alpha,\beta}$, which is still somewhat risky. In the future, instead of approximating a function, new activation functions will be created to make the network behave in a chaotic manner.

In conclusion, it is clear from the research conducted so far that the proposed technique can be used in a secure communication system for images, and that it has been effective and has further increased the difficulty of theft. However, the need to improve on the details to make the technique more perfect is definitely a future issue.

Author Contributions: Conceptualization, H.S.; methodology, Y.C., H.S. and T.Y.; formal analysis, Y.C., H.S., M.W. and T.Y.; data curation, Y.C. and T.Y.; writing—original draft preparation, Y.C., H.S., M.W. and T.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This study is supported by JSPS KAKENHI Grant Number JP 21K03370 and JST CREST JPMJCR1914.

Institutional Review Board Statement: Not applicatble.

Informed Consent Statement: Not applicable.

Data Availability Statement: The details of the data are described in Sections 2.4 and 3.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cuomo, K.M.; Oppenheim, A.V. Circuit Implementation of Synchronized Chaos with Applications to Communications. *Phys. Rev. Lett.* **1993**, *1*, 65–68. [CrossRef]
2. Kocarev, L.; Parlitz, U. General Approach for Chaotic Synchronization with Applications to Communication. *Phys. Rev. Lett.* **1995**, *25*, 5028–5031. [CrossRef]
3. Lin, C.H.; Hu, G.H.; Chan, C.Y.; Yan, J.J. Chaos-Based Synchronized Dynamic Keys and Their Application to Image Encryption with an Improved AES Algorithm. *Appl. Sci.* **2021**, *11*, 1329. [CrossRef]
4. Ushio, T. Chaotically Synchronizing Control and Its Application to Secure Communication. *Trans. Inf. Process. Soc. Jpn.* **1995**, *3*, 525–530.
5. Yoshimura, K. Multichannel Digital Communications by the Synchronization of Globally Coupled Chaotic Systems. *Phys. Rev. E* **1999**, *2*, 1648–1657. [CrossRef]
6. Li, C.; Zhao, F.; Liu, C.; Lei, L.; Zhang, J. A Hyperchaotic Color Image Encryption Algorithm and Security Analysis. *Secur. Commun. Netw.* **2019**, *2019*, 8132547. [CrossRef]
7. Sbiaa, F.; Kotel, S.; Zeghid, M.; Tourki, R.; Machhout, M.; Baganne, A. High-Level Implementation of a Chaotic and AES Based Crypto-System. *J. Circuits, Syst. Comput.* **2017**, *26*, 1750122. [CrossRef]
8. Arab, A.; Rostami, M.J.; Ghavami, B. An image encryption method based on chaos system and AES algorithm. *J. Supercomput.* **2019**, *75*, 6663–6682. [CrossRef]
9. Suri, S.; Vijay, R. An AES–CHAOS-Based Hybrid Approach to Encrypt Multiple Images. In *Recent Developments in Intelligent Computing, Communication and Devices*; Springer: Singapore, 2017; pp. 3–43.
10. Liu, Y.; Tong, X.; Hu, S. A family of new complex number chaotic maps based image encryption algorithm. *Signal Process. Image Commun.* **2013**, *28*, 1548–1559. [CrossRef]
11. Huang, X.; Ye, G. An image encryption algorithm based on hyper-chaos system and DNA plane. *Multimed. Tools Appl.* **2014**, *72*, 57–70. [CrossRef]
12. Kuo, C.L. Design of a fuzzy sliding-mode synchronization controller for two different chaos systems. *Comput. Math. Appl.* **2011**, *61*, 2090–2095. [CrossRef]
13. Moon, S.; Baik, J.-J.; Seo, J.M. Chaos synchronization in generalized Lorenz systems and an application to image encryption. *Commun. Nonlinear Sci. Numer. Simul.* **2021**, *96*, 105708. [CrossRef]
14. Njah, A.N. Tracking control and synchronization of the new hyperchaotic Liu system via backstepping techniques. *Nonlinear Dyn.* **2010**, *61*, 1–9. [CrossRef]
15. Pai, M.C. Global synchronization of uncertain chaotic systems via discrete-time sliding mode control. *Appl. Math. Comput.* **2014**, *228*, 663–671. [CrossRef]
16. Yau, H.T.; Kuo, C.L.; Yan, J.J. Fuzzy sliding mode control for a class of chaos synchronization with uncertainties. *Int. J. Nonlinear Sci. Numer. Simul.* **2006**, *7*, 333–338. [CrossRef]
17. Yu, Y.; Li, H.X. Adaptive hybrid projective synchronization of uncertain chaotic systems based on backstepping design. *Nonlinear Anal. Real World Appl.* **2011**, *12*, 388–393. [CrossRef]
18. Pecora, L.M.; Carroll, T.L. Synchronization of chaotic systems. *Chaos* **2015**, *25*, 097611. [CrossRef] [PubMed]
19. Sano, H.; Wakaiki, M.; Yaguchi, T. Secure Communication Systems Using Distributed Parameter Chaotic Synchronization. *Trans. Soc. Instrum. Control. Eng.* **2021**, *2*, 78–85. [CrossRef]
20. Chen, G.; Hsu, S.B.; Zhou, J. Chaotic Vibrations of the One-Dimensional Wave Equation due to a Self-Excitation Boundary Condition. Part I: Controlled Hysteresis, Appendix C by G.R. Chen and G. Crosta. *Trans. Am. Math. Soc.* **1998**, *11*, 4265–4311. [CrossRef]
21. Chen, G.; Hsu, S.B.; Zhou, J. Chaotic Vibration of the Wave Equation with Nonlinear Feedback Boundary Control: Progress and Open Questions. In *Chaos Control: Theory and Applications*; Chen, G.R., Yu, X., Eds.; Springer: Berlin/Heidelberg, Germany, 2003; pp. 25–50.
22. Li, L.; Huang, Y.; Xiao, M. Observer Design for Wave Equations with Van Der Pol Type Boundary Conditions. *SIAM J. Control. Optim.* **2012**, *3*, 1200–1219. [CrossRef]
23. Naoe, H.K. Tanaka and Y. Takefuji, Information Security Techniques Based on Artificial Neural Network. *J. Jpn. Soc. Artif. Intell.* **1995**, *5*, 577–585.
24. Tariq, M.I.; Memon, N.A.; Ahmed, S. A Review of Deep Learning Security and Privacy Defensive Techniques. *Mob. Inf. Syst.* **2020**, *2020*, 6535834. [CrossRef]
25. Evans, L.C. *Partial Differential Equations*; Amer Mathematical Society: Providence, RI, USA, 2010.
26. Guan, K. Important Notes on Lyapunov Exponents. *arXiv* **2014**, arXiv:1401.3315.
27. Inoue, Y. Chaos:Definition and Characterization. *Jpn. J. Multiph. Flow* **1997**, *2*, 157–162. [CrossRef]
28. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
29. Preishuber, M.; Hütter, T.; Katzenbeisser, S.; Uhl, A. Depreciating Motivation and Empirical Security Analysis of Chaos-Based Image and Video Encryption. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 2137–2150. [CrossRef]

30. Wu, Y.; Noonan, J.P.; Agaian, S. NPCR and UACI Randomness Tests for Image Encryption. *Cyber J. Multidiscip. J. Sci. Technol.* **2011**, *1*, 31–38.
31. Ali, F.; Mohammed, A.H. Content Based Image Retrieval (CBIR) by Statistical Methods. *Baghdad Sci. J.* **2020**, *17*, 694–700. [CrossRef]
32. Seetharaman, K.; Jeyakarthic, M. Statistical distributional approach for scale and rotation invariant color image retrieval using multivariate parametric tests and orthogonality condition. *J. Vis. Commun. Image Represent.* **2014**, *25*, 727–739. [CrossRef]