厦門大學

本 科 毕 业 论 文 （设 计）

（主修专业）

模板示例：基于混沌学理论的数字图像加密技术

Template Example: Digital Image Encryption Technology Based on Chaos Theory

姓　　名：张三

学　　号：22920190000000

学　　院：信息学院

专　　业：人工智能

年　　级：2019 级

校内指导教师：张三　教授

校外指导教师：张三　经理

二〇二二年五月二十六日

厦门大学本科学位论文诚信承诺书

本人呈交的学位论文是在导师指导下独立完成的研究成果。本人在论文写作中参考其他个人或集体已经发表的研究成果，均在文中以适当方式明确标明，并符合相关法律规范及《厦门大学本科毕业论文（设计）规范》。

该学位论文为（数字图像处理）课题（组）的研究成果，获得（数字图像处理）课题（组）经费或实验室的资助，在（人工智能）实验室完成。

本人承诺辅修专业毕业论文（设计）（如有）的内容与主修专业不存在相同与相近情况。

学生声明（签名）：

年　　月　　日

致　谢

致谢对象包括资助研究工作的基金、组织或个人，协助完成研究工作的组织或个人，在研究工作中提出重要建议或提供重要帮助的组织或个人以及给予转载和引用权的资料、图片、文献、研究思想和设想的所有者等。

致谢限一页。标题和篇眉内容均为“致谢”。

示例：

衷心感谢导师张三教授和人工智能系李四副教授对本人的精心指导。他们的言传身教将使我终生受益。

在美国麻省理工学院化学系进行九个月的合作研究期间，承蒙 Robert Field 教授热心指导与帮助，不胜感激。

感谢人工智能实验室主任王五教授，以及实验室全体老师和同窗们学的热情帮助和支持！

本课题承蒙国家自然科学基金资助，特此致谢。

摘 要

论文的摘要是对论文研究内容和成果的高度概括。摘要应对论文所研究的问题及其研究目的进行描述，对研究方法和过程进行简单介绍，对研究成果和所得结论进行概括。

摘要应具有独立性和自明性，其内容应包含与论文全文同等量的主要信息。使读者即使不阅读全文，通过摘要就能了解论文的总体内容和主要成果。

论文摘要的书写应力求精确、简明。切忌写成对论文书写内容进行提要的形式，尤其要避免“第 1 章……；第 2 章……；……”这种或类似的陈述方式。

示例：

近几十年来，图像加密作为重要的信息安全领域之一，吸引了众多研究人员和科学家。然而，已经用不同的方法进行了几项研究，并且已经提出了新颖且有用的算法来改进安全图像加密方案。如今，混沌方法已在多个领域中被发现，例如密码系统的设计和图像加密。基于混沌方法的数字图像加密是一种新颖的图像加密方法。该技术采用随机混沌序列对图像进行加密，是一种高度安全、快速的图像加密方法。有限的准确性是这种技术的缺点之一。本文通过研究混沌序列和小波变换值来寻找差距。因此，提出了一种用于数字图像加密的新技术并改进了以前的算法。该技术在 MATLAB 中运行，并根据像素数变化率 (NPCR)、峰值信噪比 (PSNR)、相关系数和统一平均变化强度 (UACI) 等各种性能指标进行比较。仿真和理论分析表明了该方案的有效性，表明该技术是实际图像加密的合适选择。

关键词：数字图像加密；图像处理；混沌随机序列；离散小波变换

Abstract

In recent decades, image encryption, as one of the significant information security fields, has attracted many researchers and scientists. However, several studies have been performed with different methods, and novel and useful algorithms have been suggested to improve secure image encryption schemes. Nowadays, chaotic methods have been found in diverse fields, such as the design of cryptosystems and image encryption. Chaotic methods-based digital image encryptions are a novel image encryption method. This technique uses random chaos sequences for encrypting images, and it is a highly-secured and fast method for image encryption. Limited accuracy is one of the disadvantages of this technique. This paper researches the chaos sequence and wavelet transform value to find gaps. Thus, a novel technique was proposed for digital image encryption and improved previous algorithms. The technique is run in MATLAB, and a comparison is made in terms of various performance metrics such as the Number of Pixels Change Rate (NPCR), Peak Signal to Noise Ratio (PSNR), Correlation coefficient, and Unified Average Changing Intensity (UACI). The simulation and theoretical analysis indicate the proposed scheme's effectiveness and show that this technique is a suitable choice for actual image encryption.

Keywords: digital image encryption; image processing; chaos random sequence; discrete wavelet transform

目　录

Table of Contents

1 引言

近年来，图像加密一直是一个有吸引力的研究领域。它被广泛认为是一种用于安全传输的有用技术。每个图像加密算法都旨在生成具有最高质量的嘈杂图像以保密信息。此外，图像加密对于保证网络上的分类传输和图像容量具有较好的作用。随着互联网技术的飞速发展，数字通信变得更加广泛。人们可以随时随地在互联网上发送数字图像。这导致了数字图像加密的发展。研究中表示数字图像加密的不同方法与日益增加的安全必要性有关。基于混沌方法的图像加密是一种新颖的图像加密方法，它采用随机混沌序列对图像进行加密，是解决高安全性和快速图像加密的棘手问题的有效途径。在过去的几年里，出现了各种版本的混沌技术。目前，已采用四种方法进行图像加密，分别应用各种原理并实现相同的目标。这四项原则包括共享和秘密分割、顺序排列、混沌动态系统和现代密码学，每项原则都具有独特的特征。[1]

本文的主要目的是提供一种基于混沌理论的数字图像加密新技术。它由 Y. Pourasad 等人提出。然而，基于混沌的图像加密技术存在一些问题，包括准确性有限。为此，在本研究中，图像的加密分为空间加密和变换域加密。在过去的几年里，一些图像加密方案被提出了频域和空间域。空间域方法直接作用于普通图像的像素。[2-3] 由于这种方法包含高速加密，因此被广泛使用。使用变换域加密，考虑到数字图像的一些典型属性，即高冗余和附近像素之间的强相关性。

本文以混沌序列和小波变换值以及图像加密算法的融合为导向。这种算法是通过分析算法来模拟的，以发现差距。因此，算法得到了增强。该方法使用两个一维混沌系统，甚至可以使用基本非线性方程来显示混沌行为。我们的主要目标，以及采用这种映射的比例，是发现一个新的离散时间序列，与具有唯一参数的基本方程的逻辑映射的混沌输出相同。本文将在以下部分中介绍。在“引言”部分，描述了问题的动机和陈述。此外，在“方法和材料”中，介绍了该方法的基本数学概念和表达式。此外，在“提议的算法”部分，使用图形和表格描述了提议的模型实现的结果。此外，比较在“算法”部分进行了介绍。最后，“结论”部分通过数值结果和透视概念总结了结果。[2,4]

2 材料与方法

本章将介绍一些基于混沌理论的数字图像加密技术的理论知识，包括混沌理论、混沌随机序列和小波变换。

2.1 混沌学概述

混沌理论最早是由麻省理工学院的数学家和气象学家 Edward Lorenz 在 60 年代初的天气预报实验中发现的。该理论即将探索明显随机数据中的隐藏模式。它提供了一种方便的方法来解决自然和人工系统的非线性问题，这些系统具有不可预测的行为，例如道路交通、股票市场、地震、健康心脏的节律、DNA 编码序列、天气和气候条件。对初始条件高度敏感的系统可以在混沌理论的保护伞下进行研究，混沌理论有意提及蝴蝶效应。蝴蝶效应通常被解释为蝴蝶在巴西拍打翅膀并在德克萨斯州引发飓风。这意味着大系统中的微小变化可能会产生复杂的结果。在这种情况下，该系统可能是从天气模式到小行星运动或人们的互动的任何东西，整个系统受到影响的微小变化。从科学上讲，它被称为对初始条件 Dooley 的敏感依赖。由于计算中的一些数值错误，产生了各种初始条件。这些误差为某些动态系统提供了大相径庭的结果。这使得几乎不可能预测长期渲染的行为。即使系统的行为是由同一系统的初始条件决定的，并且过程中不涉及随机元素，也会发生这种情况。具有这种条件的动态系统称为确定性系统。不足以使它们可预测的这种确定性行为的动态系统被标记为确定性。因此，Edward Lorenz 尝试用一个单一的定义来描述混沌理论的主要概念。他说：“现在可以决定未来，但大概的现在不能决定大概的未来”。这种预测随机性问题是一个巨大的问题。

2.2 基于逻辑图的混沌序列

显示二次非线性的一个离散时间维非线性系统称为逻辑图。逻辑图函数可以表示为：

$$f(x) = \mu_x(1 - x) \tag{2.1}$$

状态方程的形式表示为:

$$x_{n+1} = f(x_n) = \mu_{x_n}(1 - x_n) \tag{2.2}$$

其中 $x_n \in (0, 1)$ 和 $\mu \in (0, 4)$ 被称为控制参数或分岔参数。

这里，x_n 表示系统在 n 时间的状态。x_{n+1} 表示下一时刻状态，n 表示离散时间。通过重复迭代关闭，增加了一系列点 $\{x_n\}_\infty$，称为轨道。逻辑图的性能对 μ 的值很敏感。对于 $\mu = 3.574$，逻辑图是混沌的。对于不同的初级条件，使用两个逻辑图来执行重复操作。此外，动态测量两个逻辑图的状态值。通过这个操作，产生了混沌序列。

2.3 小波变换

用于分析信号频率分量的一种有价值的仪器称为傅里叶变换。以傅立叶在整个时间轴上进行转换，不可能确定增加特定频率的确切时刻。傅里叶变换和小波变换是相同的，具有

完全不同的评价函数。小波变换主要旨在仅通过变换时间扩展而不是形状来允许变化。两者的主要区别在于信号通过傅里叶变换分解为余弦和正弦；然而，小波变换利用了傅里叶和实空间中的函数。通常，小波变换表述如下：

$$f(a,b) = \int_{\infty}^{-\infty} f(x)\Psi^{*}(a,b)\,\mathrm{d}x \tag{2.3}$$

其中 * 代表复共轭符号，函数是一个函数，只要它遵循一定的规则，就可以任意选择。小波变换可以将信号转换为时间、空间和频率作为独立的空间。它还侧重于任何局部细节的特定信号。因此，通过小波变换可以有效地从信号中提取更多信息。

存在多种类型的小波变换用于特定目的。我们使用连续和离散小波变换从信号中提取更多信息。与傅里叶变换类似，连续小波变换使用内积来测量信号和分析函数之间的相似性。理论分析是使用连续小波变换的领域之一。在作为研究功能领域的计算机的特定实现中，必须对连续小波进行离散化。按照一些确定的规则通过一组离散的小波尺度和平移运行小波变换称为离散小波变换。信号通过变换为相互正交的小波群来分解，作为连续小波变换的必要变化。此外，离散时间序列的实现有时被确定为离散时间连续小波变换。选择用于时频分解的小波是最重要的一点。通过这种选择，我们可以影响结果的频率和时间分辨率。这种方式不能替代小波变换（WT）的基本特征（低频具有错误的时间分辨率和真实频率；较高频率具有错误的频率分辨率和良好的时间）。然而，以某种方式增加总时间分辨率的总频率是可能的。它与傅里叶和实空间中使用的小波宽度成正比。使用 Morlet 小波，我们可以假设高频分辨率是频率中非常好的局部化小波。相反，利用高斯小波的导数将导致正确的时间定位但频率较低。

3 算法

本节可按小标题划分。它应该对实验结果、它们的解释以及可以得出的实验结论提供简明准确的描述。实施建议算法的步骤是：

步骤 1：排列一张灰度图。图像的大小设置为 $m \times n$。此外，放置了数据矩阵 R。通过评估两个逻辑图，生成一个混沌序列。与主图像进行 XNOR，扩散终止。

步骤 2：本步骤中，对步骤 1 中的漫反射图像进行小波分解，提取小波系数，记为 ca1。

步骤 3：利用二维超混沌图 CML 产生混沌序列，并利用步骤 2 中建立的 ca1 进行位置混淆。

步骤 4：在最后一步，可以通过小波重建混淆图像，得到了加密的图像。加密的逆运算称为解密算法。图像加密和图像解密中的系统参数和混沌序列的初值是一致的。

3.1 加密评估指标

我们通过选择一些基本参数来评估算法来衡量我们的密码方案的性能。视觉检查是评估加密图像的主要参数之一。特征扩散调查是判断随机化算法的另一个参数。通过检查，确定产品与一组确定的特征的偏差。人工操作通常完成检查；然而，机器视觉被用于自动化这个过程。由于算法的良好扩散，原始图像和加密图像之间的关联变得过于复杂，无法简单地预测。在这里，我们研究了峰值信噪比 (PSNR) 计算指标，即加密图像和关键图像之间的关联。最终，我们通过计算统一平均变化强度 (UACI) 和像素数变化率 (NPCR) 两个参数来评估规范扩散。[5]

3.2 峰值信噪比 (PSNR)

峰值信噪比 (PSNR) 是通过均方误差 (MSE) 确定的工程公式，它通常用于图像质量评估。公式如下[6]：

$$\mathrm{PSNR} = 10\log\left(\frac{255^2}{\mathrm{MSE}(f, f')}\right) \tag{3.1}$$

其中 $f(x,y)$ 和 $f'(x,y)$ 表示原始图像和重建图像的像素值。

3.3 像素数变化率 (NPCR)

扩散以判断加密算法随机化的最基本参数的个数来表示。NPCRs 用于检查图像加密算法的安全性。考虑 C_1 和 C_2 作为两个具有大小的图像，我们定义了一个与图像大小相似的数组：

$$D(i,j) = \begin{cases} 0, \text{ if } C_1(i,j) = C_2(i,j) \\ 1, \text{ if } C_1(i,j) \neq C_2(i,j) \end{cases} \tag{3.2}$$

NPCR 确定两个不同图像中像素的百分比，其计算公式如下[2,7-8]：

$$\mathrm{NCPR} = \frac{\sum_i \sum_j D(i,j)}{N \times M} \times 100\% \tag{3.3}$$

3.4 统一平均变化强度 (UACI)

UACI 使用以下表达式确定两个加密图像 (C_1 和 C_2) 内差异的平均强度，用于评估加密方法的强度，它的值基于图像的格式和大小。通过 UACI，可以评估加密图像和原始图像之间的平均强度变化。最大的 UACI 表明所建议的技术对各种攻击具有抵抗力。UACI 的确定如下（假设灰度图像的大小为 $M \times N$）[9]：

$$\mathrm{UACI} = \frac{1}{N \times M}\left[\sum_{i}\sum_{j}\frac{C_1(i,j) - C_2(i,j)}{\max(C_2)}\right] \times 100 \tag{3.4}$$

3.5 数字图像关联 (DIC)

数字图像相关 (DIC) 是一种关键且广泛使用的非接触式方法来测量材料变形。近年来，在开发新颖的实验 DIC 方法和提高相关计算算法的性能方面取得了重大进展。因此，加密图像和原始图像的相同像素之间的关系如下所示：

$$\mathrm{NC} = \sum_{m}\sum_{n}\frac{(A_{mn} - \overline{A})(B_{mn} - \overline{B})}{\sqrt{(A_{mn} - \overline{A})^2 + (B_{mn} - \overline{B})^2}} \tag{3.5}$$

其中 A 和 B 分别表示原始图像和加密图像，以及它们的均值。较低的相关系数值是最佳的。

4　实验和数值结果

在本章中，我将使用 Y. Pourasad 等人的实验结果所提出的算法步骤的结果，如图 1 所示。在第一步中，导入大小为 $m \times n$ 的输入灰度图像。基于图4.1[①]，使用使用的两个逻辑图创建了一个混沌序列。最后，在扩散步骤中，生成用于加密的安全密钥。对于输入图像的加密，必须在小波分解子带之间插入安全密钥。DWT 方法的子带如图4.1所示。DWT 子带中从上到下和从左到右的图像是 Low-Low、Low-High、High-Low 和 High-High 子带。利用二维超混沌映射 CML，产生混沌序列并进行混淆。在最后一步，生成混淆图像。最后，图像由使用输入图像和安全密钥的加密矩阵组成。用数值结果评估建议的算法表明该算法是稳健的。所提出算法的数值结果如表4.1[②]所示。

表 4.1　所提出的算法的数值结果

Image	Type	PSNR	NPCR	UACI	NC
Lena	JPG	42.612	99.757	33.120	0.9548
Peppers	JPG	39.220	99.787	33.621	0.9934
Barbara	JPG	36.841	99.626	33.126	0.9809
Baloon	JPG	39.134	99.881	33.415	0.9137
Boat	JPG	39.223	99.625	33.671	0.9001

首先，对原始图像（原始图像）进行扩散操作，主键值取自表4.2。然后，在从表4.2中获取主键值的同时执行混淆操作。加密的结果与噪声相同（图4.2）。从加密图像中没有获取关于原始图像的信息。通过对加密图像进行解密的密钥获取解密图像，然后进行扩散和混淆操作（图4.2(b)）。

表 4.2　扩散和混淆操作的关键初始值

Key (Diffusion)	Value (Diffusion)	Key (Confusion)	Value (Confusion)
$x_1(1)$	0.5	$x_3(1)$	0.3
$x_2(1)$	0.5	$y_3(1)$	0.3
μ_1	4.0	μ_1	4.0
μ_2	3.9	μ_2	3.9

4.1　直方图分析

第一个测试是加密、解密和原始图像的直方图分析。在这里，各个图像的图像直方图代表了加密图像和原始图像之间的巨大差异，但它们是相同的。通过对测试图像的直方图和加密后的直方图的评价，可以观察到加密后的图像在直方图的整个区间内是均匀分布的。因此，覆盖了原始图像的分布规律。因此，有效地实施了加密（参见图4.3）。

① （顺带当作是脚注示例）注意给图片标签命名时不要像我这样使用 4.1，不然如果你要在前面加图片会导致很混乱，最好起别的名字

② 注意给表格标签命名时不要像我这样使用 4.1，不然如果你要在前面加表格会导致很混乱，最好起别的名字（同样顺带当作是脚注示例）

4.2 鲁棒性 (健壮性)

除了图4.4 中的直方图分析之外，Y. Pourasad 等人还评估了输入图像和加密图像中两个垂直、两个水平和两个对角相邻像素之间的相关性。图像中两个相邻像素的值由 x 轴和 y 轴表示。在输入图像和密码图像中，图4.4描绘了两个水平相邻像素的相关分布。普通图像和密码图像的相关系数分别为 0.99 和 0.02。对角线和垂直方向都产生相似的结果。简单的图片具有两个相邻像素的高度相关性。

为了评估所提出的方法的鲁棒性，测试图像针对四种类型的图像处理攻击进行了测试：旋转、高斯噪声、中值过滤和直方图均衡。结果表明，所提出的设计具有更高的鲁棒性和归一化相关性。根据结果，输入攻击不影响图像加解密。关于不同类型图像的归一化相关 (NC) 值，中值滤波器、旋转和高斯噪声具有更高的 NC 值。这意味着所提出的方法的鲁棒性可以抵抗这些类型的攻击。然而，直方图均衡的影响是显着的（见表4.3）。

表 4.3　所提算法在不同攻击类型下的 NC 值

Image	Median Filter	Histogram Equalization	Rotation	Gaussian Noise
Lena	0.984	0.987	0.999	0.999
Peppers	0.704	0.280	0.923	0.964
Barbara	0.914	0.497	0.980	0.991
Baloon	0.960	0.629	0.991	0.996
Boat	0.976	0.746	0.995	0.998

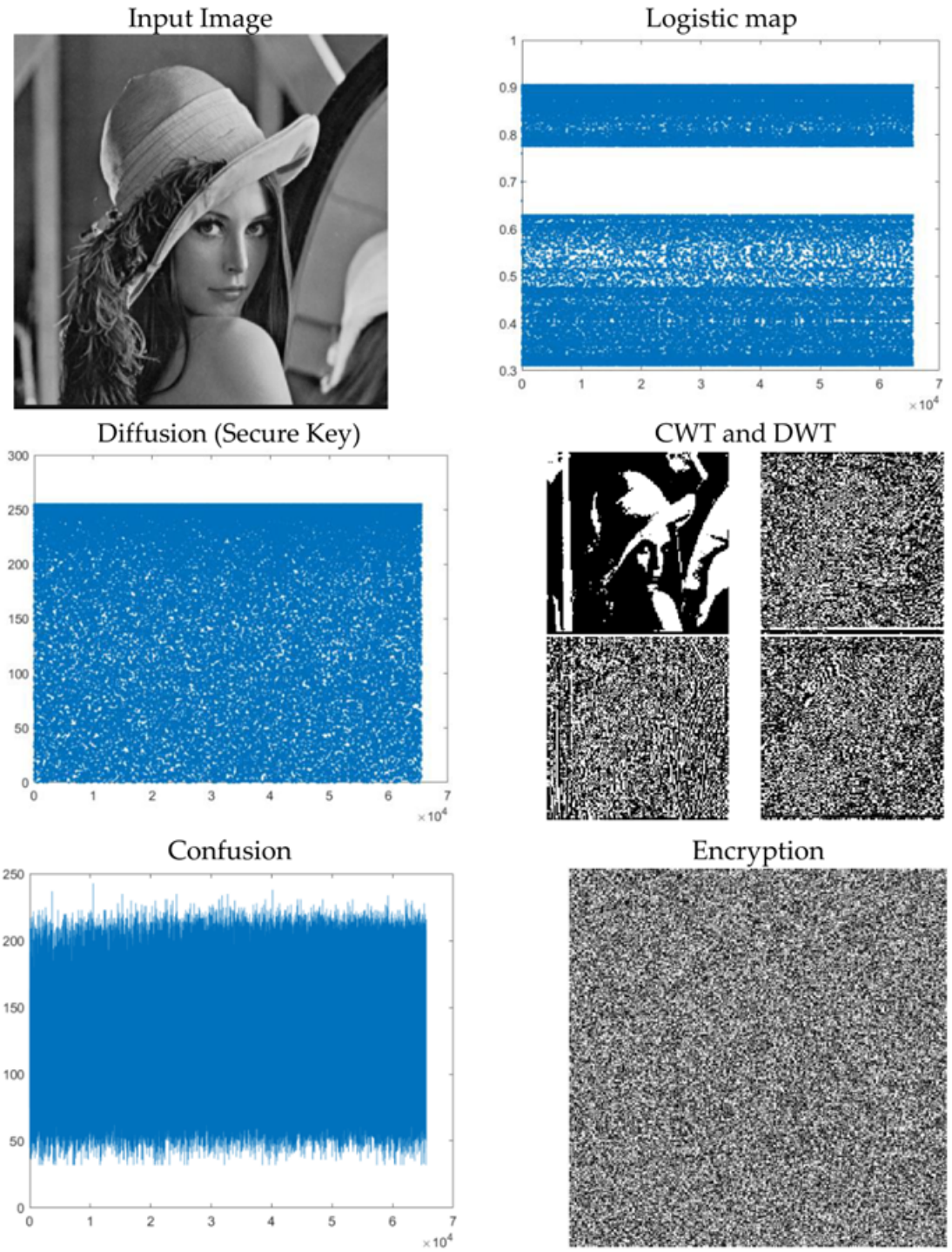

图 4.1　所提出方法的处理结果

图 4.2　将所提出的算法应用于某些图像的视觉结果
(a) 原始图像；(b) 加密图像；(c) 重建图像

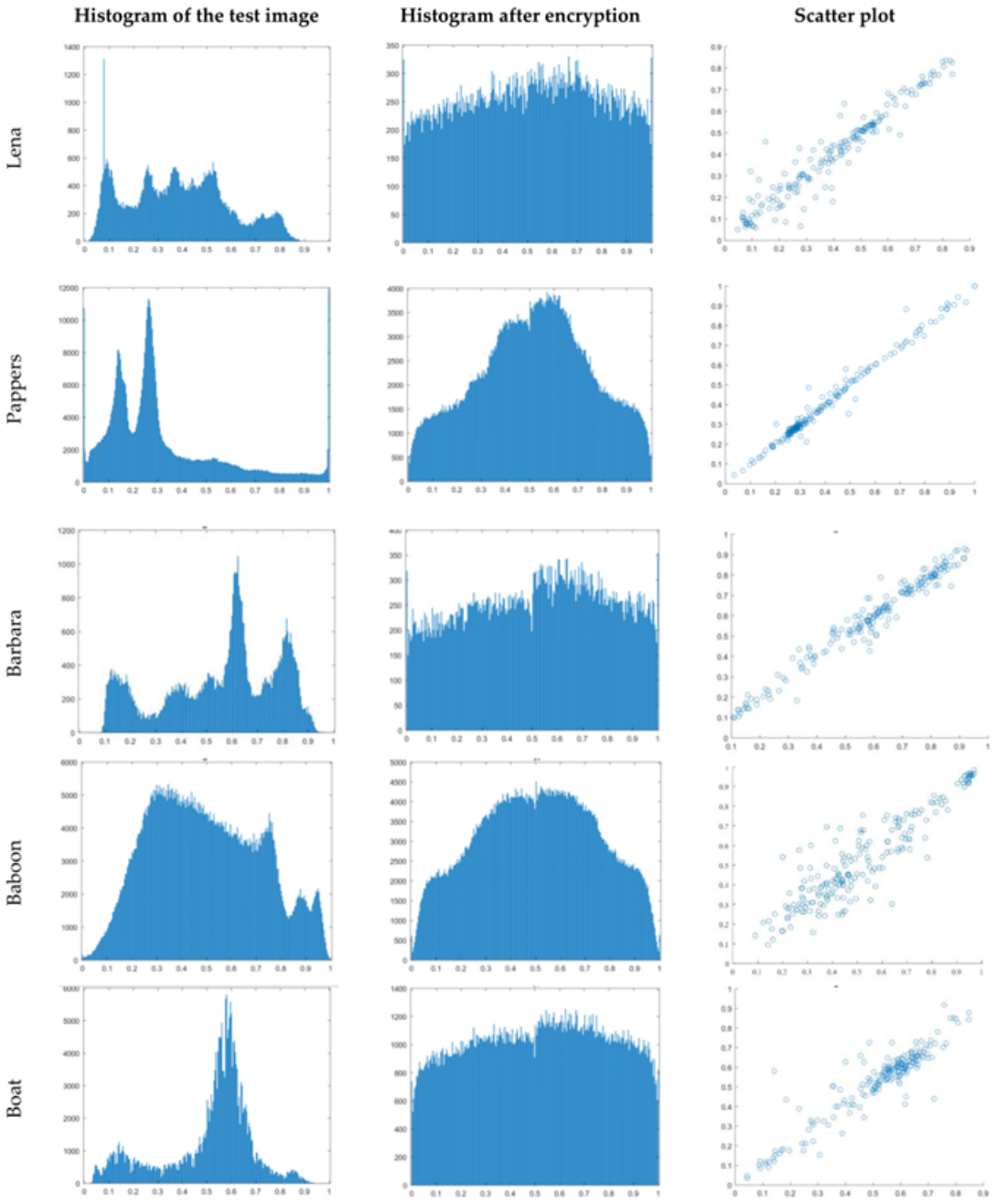

图 4.3　原始 Lena 图像的直方图分析

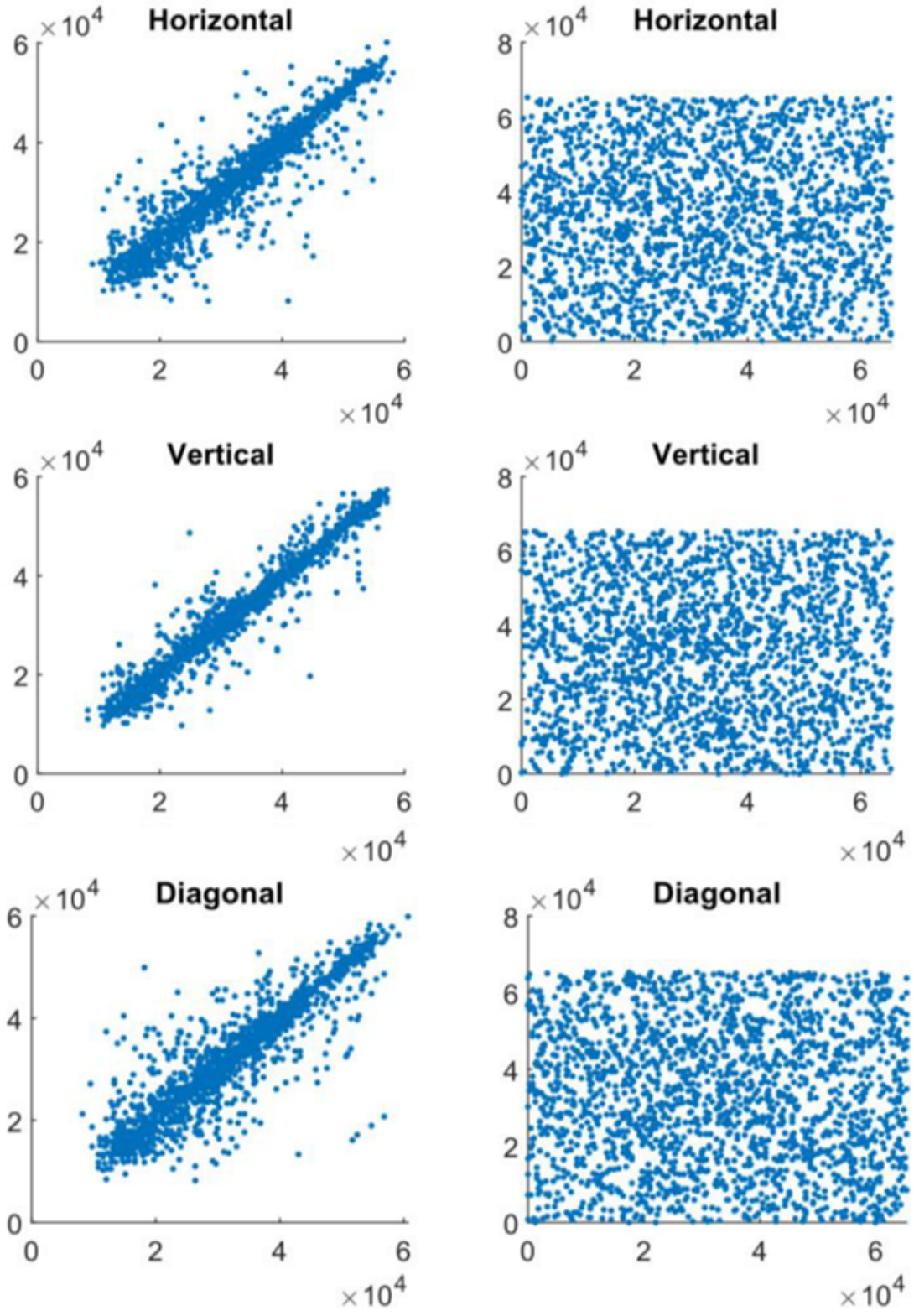

图 4.4　Lena 图像中的像素水平、垂直和对角线、输入（左）和加密图像（右）的相关性图

5 结论

最近，已经提出了各种基于混沌的图像密码系统。目前的工作是利用混沌映射和小波变换的特征来处理基于混沌的算法该算法的加密过程包括两个阶段。首先，我们进行了图像扩散操作。此外，通过执行小波变换，由于超混沌序列而大大减少了混淆计算量。标准度量的仿真结果表明，所提出的算法对密钥具有高度的依赖性。该算法包括一个不错的加密效果。此外，它可以抵抗噪音和减少攻击。Y. Pourasad 等人已经从基准 MATLAB 测试图像中测试了 Lena、Peppers、Barbara、Baboon 和 Boat 图像的提出方法。此外，还描述了输入图像和加密图像的直方图。此外，还记录了 PSNR、NPCR、UACI 和 NC 等加密性能分析标准。根据结果，Lena、Peppers、Barbara、Baboon 和 Boat 的相关值分别为 95.48%、99.64%、98.09%、91.37% 和 90.01%。为了评估所提出的方法的鲁棒性，测试图像针对四种类型的图像处理攻击进行了测试：旋转、高斯噪声、中值过滤和直方图均衡。结果表明，所提出的设计具有更高的鲁棒性和归一化相关性。根据结果，输入攻击不影响图像加解密。关于不同类型图像的 NC 值，中值滤波、旋转和高斯噪声具有更高的 NC 值。这意味着所提出的方法的鲁棒性可以抵抗这些类型的攻击。然而，直方图均衡的影响是显著的。

参考文献

[1] FU C, CHEN J J, ZOU H, et al. A chaos-based digital image encryption scheme with an improved diffusion strategy[J]. Optics express, 2012, 20(3): 2363-2378.

[2] YUN-PENG Z, WEI L, SHUI-PING C, et al. Digital image encryption algorithm based on chaos and improved des[C]//2009 IEEE international conference on systems, man and cybernetics. IEEE, 2009: 474-479.

[3] WANG Q, DING Q, ZHANG Z, et al. Digital image encryption research based on dwt and chaos[C]//2008 Fourth International Conference on Natural Computation: volume 5. IEEE, 2008: 494-498.

[4] POURASAD Y, RANJBARZADEH R, MARDANI A. A new algorithm for digital image encryption based on chaos theory[J]. Entropy, 2021, 23(3): 341.

[5] UL HAQ T, SHAH T. Algebra-chaos amalgam and dna transform based multiple digital image encryption[J]. Journal of Information Security and Applications, 2020, 54: 102592.

[6] XIAO D, LIAO X, WEI P. Analysis and improvement of a chaos-based image encryption algorithm[J]. Chaos, Solitons & Fractals, 2009, 40(5): 2191-2199.

[7] WANG S, WANG C, XU C. An image encryption algorithm based on a hidden attractor chaos system and the knuth–durstenfeld algorithm[J]. Optics and Lasers in Engineering, 2020, 128: 105995.

[8] WANG W, TAN H, SUN P, et al. A novel digital image encryption algorithm based on wavelet transform and multi-chaos[C]//Wireless communication and sensor network: proceedings of the international conference on wireless communication and sensor network (WCSN 2015). World Scientific, 2016: 711-719.

[9] YING W, DELING Z, LEI J. Digital image encryption algorithm based on three-dimension lorenz chaos system[J]. 工程科学学报, 2004, 26(6): 678-682.

附录 A　关于本论文模板的相关说明

此论文仅作模板示例用，有关内容的正确与否不做研究，参考文献引用部分也仅做演示用，请勿引用或直接抄录模板里的示例内容！否则后果自负。

本科毕业论文规范见厦门大学教务处文件：

https://jwc.xmu.edu.cn/2016/0506/c2160a173611/page.htm。

本论文模板编写参考书籍：《LaTex 入门》（刘海洋著）。

附录 B　附录代码示例

注意！代码中最好不要出现中文（包括注释），如果确实需要的，请自行搜索如何兼容中文。示例：

模型训练的代码如下：

```
"""
File: train.py
Author: San Zhang
Date: 2022-04-12
Description: Use the MindSpore framework to train a ResNet50-based cat and
dog classification neural network model.
"""

import os
import stat
import numpy as np
import matplotlib.pyplot as plt
import mindspore.nn as nn
import mindspore.dataset as ds
import mindspore.dataset.vision.c_transforms as CV
import mindspore.dataset.transforms.c_transforms as C
from mindspore import dtype as mstype
from mindspore.train.callback import TimeMonitor, Callback
from mindspore import Model, Tensor, context, save_checkpoint, \
    load_checkpoint, load_param_into_net
from resnet import resnet50

# Set use CPU/GPU/Ascend
context.set_context(mode=context.GRAPH_MODE, device_target="CPU")

# Set data path
train_data_path = 'dataset/train'
val_data_path = 'dataset/val'


def create_dataset(data_path, batch_size=100, repeat_num=1):
    data_set = ds.ImageFolderDataset(data_path, num_parallel_workers=2,
                                     shuffle=True)

    image_size = [224, 224]
    mean = [0.485 * 255, 0.456 * 255, 0.406 * 255]
    std = [0.229 * 255, 0.224 * 255, 0.225 * 255]
    trans = [
        CV.Decode(),
        CV.Resize(image_size),
        CV.Normalize(mean=mean, std=std),
        CV.HWC2CHW()
    ]
```

```
44
45      type_cast_op = C.TypeCast(mstype.int32)
46      data_set = data_set.map(operations=trans, input_columns="image",
47                              num_parallel_workers=2)
48      data_set = data_set.map(operations=type_cast_op, input_columns="label",
49                              num_parallel_workers=2)
50      data_set = data_set.batch(batch_size, drop_remainder=True)
51      data_set = data_set.repeat(repeat_num)
52
53      return data_set
54
55
56  train_ds = create_dataset(train_data_path)
57
58
59  def apply_eval(eval_param):
60      eval_model = eval_param['model']
61      eval_ds = eval_param['dataset']
62      metrics_name = eval_param['metrics_name']
63      res = eval_model.eval(eval_ds)
64      return res[metrics_name]
65
66
67  class EvalCallBack(Callback):
68
69      def __init__(self, eval_function, eval_param_dict, interval=1,
70                   eval_start_epoch=1, save_best_ckpt=True,
71                   ckpt_directory="./", besk_ckpt_name="best.ckpt",
72                   metrics_name="acc"):
73          super(EvalCallBack, self).__init__()
74          self.eval_param_dict = eval_param_dict
75          self.eval_function = eval_function
76          self.eval_start_epoch = eval_start_epoch
77          if interval < 1:
78              raise ValueError("interval should >= 1.")
79          self.interval = interval
80          self.save_best_ckpt = save_best_ckpt
81          self.best_res = 0
82          self.best_epoch = 0
83          if not os.path.isdir(ckpt_directory):
84              os.makedirs(ckpt_directory)
85          self.best_ckpt_path = os.path.join(ckpt_directory,
86                                             besk_ckpt_name)
87          self.metrics_name = metrics_name
88
89      def remove_ckpoint_file(self, file_name):
90          os.chmod(file_name, stat.S_IWRITE)
91          os.remove(file_name)
92
```

```
    def epoch_end(self, run_context):
        cb_params = run_context.original_args()
        cur_epoch = cb_params.cur_epoch_num
        loss_epoch = cb_params.net_outputs
        if cur_epoch >= self.eval_start_epoch and \
                (cur_epoch - self.eval_start_epoch) % self.interval == 0:
            res = self.eval_function(self.eval_param_dict)
            print('Epoch {}/{}'.format(cur_epoch, num_epochs))
            print('-' * 10)
            print('train Loss: {}'.format(loss_epoch))
            print('val Acc: {}'.format(res))
            if res >= self.best_res:
                self.best_res = res
                self.best_epoch = cur_epoch
                if self.save_best_ckpt:
                    if os.path.exists(self.best_ckpt_path):
                        self.remove_ckpoint_file(self.best_ckpt_path)
                    save_checkpoint(cb_params.train_network,
                                    self.best_ckpt_path)

    def end(self, run_context):
        print("End training, the best {0} is: {1}, "
              "the best {0} epoch is {2}".format(self.metrics_name,
                                                 self.best_res,
                                                 self.best_epoch),
              flush=True)


def visualize_model(best_ckpt_path,val_ds):
    net = resnet50(2)
    param_dict = load_checkpoint(best_ckpt_path)
    load_param_into_net(net,param_dict)
    loss = nn.SoftmaxCrossEntropyWithLogits(sparse=True,reduction='mean')
    model = Model(net, loss,metrics={"Accuracy":nn.Accuracy()})
    data = next(val_ds.create_dict_iterator())
    images = data["image"].asnumpy()
    labels = data["label"].asnumpy()
    class_name = {0:"cat",1:"dog"}
    output = model.predict(Tensor(data['image']))
    pred = np.argmax(output.asnumpy(), axis=1)

    plt.figure(figsize=(12,5))
    for i in range(24):
        plt.subplot(3,8,i+1)
        color = 'blue' if pred[i] == labels[i] else 'red'
        plt.title('pre:{}'.format(class_name[pred[i]]), color=color)
        picture_show = np.transpose(images[i],(1,2,0))
        picture_show = picture_show/np.amax(picture_show)
        picture_show = np.clip(picture_show, 0, 1)
```

```
        plt.imshow(picture_show)
        plt.axis('off')
    plt.show()


def filter_checkpoint_parameter_by_list(origin_dict, param_filter):
    for key in list(origin_dict.keys()):
        for name in param_filter:
            if name in key:
                print("Delete parameter from checkpoint: ", key)
                del origin_dict[key]
                break


# Define Network
net = resnet50(2)
num_epochs=5

# Define optimizer and loss function
opt = nn.Momentum(params=net.trainable_params(),
                  learning_rate=0.1, momentum=0.9)
loss = nn.SoftmaxCrossEntropyWithLogits(sparse=True, reduction='mean')

# Instantiate model
model = Model(net, loss, opt, metrics={"Accuracy": nn.Accuracy()})

# Load traing dataset and evaluation dataset
train_ds = create_dataset(train_data_path)
val_ds = create_dataset(val_data_path)

# Instantiate the callback class
eval_param_dict = {"model": model,"dataset": val_ds,
                   "metrics_name": "Accuracy"}
eval_cb = EvalCallBack(apply_eval, eval_param_dict,)

# Training model
model.train(num_epochs,train_ds,
            callbacks=[eval_cb, TimeMonitor()], dataset_sink_mode=False)

visualize_model('best.ckpt', val_ds)
```