2021 年 2 月 第 2 期
Vol. 42 No. 2 2021

一种使用机器学习方法的数字水印算法

孙晓霞,佟国香

(上海理工大学 光电信息与计算机工程学院,上海 200093)
E-mail: 2386419235@qq.com

摘 要: 针对目前数字水印算法存在的不足,本文将离散小波变换和奇异值分解相结合,提出了一种基于机器学习的图像数字水印算法. 首先将载体图像进行一级小波变换,提取其低频子带图像对其进行 4×4 分块处理,然后对每一分块进行奇异值分解后嵌入水印,并提取特征向量用于最小二乘支持向量机的训练,训练好的最小二乘支持向量机用于自适应最大水印嵌入强度的计算以及水印的盲提取. 实验选取三张 512×512 的标准测试图像以及 64×64 的二值水印图像对算法的透明性与鲁棒性进行测试. 实验结果证明,图像具有很好的透明性,PSNR 达到了 63.71dB,针对旋转、剪切、JPEG 压缩、高斯噪声等常规攻击手段时,算法能保持较强的鲁棒性.

关 键 词: 数字水印; 机器学习; 最小二乘支持向量机; 奇异值分解; 离散小波变换
中图分类号: TP391 **文献标识码:** A **文 章 编 号:** 1000-1220(2021)02-0387-06

Digital Watermarking Algorithm Using Machine Learning Method

SUN Xiao-xia, TONG Guo-xiang
(School of Optional Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: In view of the shortcomings of current digital watermarking algorithms, this paper proposes a machine learning based image digital watermarking algorithm by combining discrete wavelet transform (DWT) and singular value decomposition (SVD). Firstly, the image is processed by DWT, and then the low-frequency sub-band image is extracted for 4×4 block processing. Then the watermark is embedded by SVD for each block, and the feature vector is extracted for the training of LS-SVM. Trained LS-SVM is used to calculate the adaptive maximum watermark embedding strength and blind watermark extraction. Three 512×512 standard test images and the 64×64 binary watermark image are selected to simulate the transparency and robustness of the algorithm. The experimental results show that the transparency of the images is very good, and the PSNR is up to 63.71dB. The algorithm can maintain strong robustness against conventional attacks such as rotation, cropping, JPEG compression, Gaussian noise and so on.

Key words: digital watermarking; machine learning; LS-SVM (Least Squares Suppore Vector Machine); SVD (Singular Value Decoposition); DWT (Discrete Wavelet Transform)

1 引 言

数字水印技术是一种应用计算机算法将保护信息嵌入载体文件的技术,对其起到版权保护及认证的作用. 随着数字信息时代的发展,数字作品的版权保护问题引起越来越多的关注. 近年来,许多国内外的专家学者在数字水印领域做出了自己的贡献,这些贡献主要集中在图像水印,音频水印,视频水印等,其中研究最广泛的是图像数字水印.

经典的图像数字水印算法包括空域数字水印算法和变换域数字水印算法[1]. 常见的空域算法有 LSB 算法[2]、Patchwork 算法、基于直方图的算法等,空域水印算法快捷简单、时间复杂度低,但是鲁棒性和透明性较差. 变换域算法将载体图像进行相应的频域变换,通过修改频域系数将水印嵌入. 与空域算法相比,变换域算法更复杂,但是不可见性和鲁棒性也更高,是目前常用的数字水印算法. 常见的变换域算法有离散余弦变换(DCT)[3]、离散傅里叶变换(DFT)和离散小波变换(DWT)[4]等.

基于 DCT 和分形编码的数字图像水印方法[5]实现了双重加密,改进了传统的 DCT 方法,提高了计算效率,但其抵抗几何攻击的能力较弱. 与 DCT 相比,小波变换抵抗几何攻击的能力更强,文献[6]提出了一种基于广义卷积定理的线性正则小波变换(LCWT)的水印算法,算法将 LCWT 和 QR 分解相结合,提供了更大的灵活性,但其水印的不可见性一般.

基于变换域的方法虽然可以提高鲁棒性,但其抵抗旋转缩放等攻击的能力整体较弱,因此一种更加稳定的变换方法奇异值分解(SVD)被引入数字水印算法. 陈青等[7]将整数小波系数与 SVD 相结合,提出了基于模运算的 SVD 小波系数算法,该方法对一般的几何攻击尤其是旋转攻击有较强的鲁棒性,但其算法复杂度较高. Makbol 等[8]将 SVD 和 DWT 相结合选取特定块区域嵌入水印,实验结果表明算法对常见的图像处理攻击具有高透明性和高鲁棒性,不足之处在于算法效率不够高.

收稿日期: 2020-03-03 收修改稿日期: 2020-04-14 基金项目: 国家重点研发计划项目(2018YFB1700902)资助. 作者简介: 孙晓霞,女,1994 年生,硕士研究生,研究方向为图像处理; 佟国香(通讯作者),女,1968 年生,博士,副教授,研究方向为嵌入式系统开发、图像处理、数据挖掘.

与一般图像不同,医学图像的图像特征通常具有重要的意义和价值,因此很多针对图像特征的生理模型算法也逐渐被提出. 文献[9]利用指纹生物识别技术进行身份验证,利用加密过程进行保密,并利用可逆水印实现完整性,提出了一种基于生物特征的有效医学图像水印的电子水印技术,但其只是针对特定领域,不适用于所有图像.

上述方法都是数字水印算法的传统方法,随着机器学习的发展,许多基于机器学习的算法被应用于图像数字水印算法领域. 基于机器学习的算法可以解决参数优化和水印盲提取两个问题[10]. 文献[11]将 DWT 和 DCT 相结合,将 SVM 回归模型用于几何畸变校正,提高了算法的鲁棒性,但是算法的复杂度很高. 基于机器学习的水印算法虽然在不同程度上实现了更好的鲁棒性与透明性,但是在参数优化以及算法复杂度等方面仍然存在很大的提升空间.

为了解决目前水印算法存在的问题,本文将 DWT 和 SVD 相结合,提出了基于机器学习的图像数字水印算法. 首先将载体图像进行一级离散小波变换,选取低频子带进行分块处理,再对分块进行奇异值分解,并提取特征向量来训练 LS-SVM 分类器. 训练好的 LS-SVM 用于自适应最大水印嵌入强度的计算以及水印的盲提取.

2 基本原理

2.1 水印预处理

将水印进行加密可以提高算法保密性,本文采用混沌加密算法对水印进行加密,混沌加密算法具有伪随机性、便利性和对初值敏感等特性,在水印加密中被广泛使用[12]. 其逻辑方程如公式(1)所示:

$$x_{n+1}=rx_n(1-x_n) \quad (r>0,0<x_i\leqslant 1) \tag{1}$$

式中:x_n 为自变量,r 为控制参数,r 取值不同时,数列$\{x_n\}$具有不同的敛散性,当 r 趋近 4 时,$\{x_n\}$呈现无规则化,出现混沌状态.

2.2 离散小波变换

离散小波变换(DWT)[13]具有多分辨率和多尺度的特性,更加符合人眼视觉系统(HVS)的特点,抵抗压缩攻击的能力更强. 将载体图像进行一级离散小波变换得到 LL,HL,LH,HH 四个子带图像,LL 子带包含了图像的主要信息,其他三个子带代表了图像的细节和边缘信息,故选取 LL 子带进行分块处理. 离散小波变换的公式如(2)所示:

$$WT(\alpha,\tau)=\frac{1}{\sqrt{\alpha}}\int_{-\infty}^{+\infty}f(t)\,\varphi\left(\frac{t-\tau}{\alpha}\right)dt \tag{2}$$

式中:α 为尺度,τ 为平移量.

2.3 奇异值分解(SVD)

奇异值分解稳定性好,具有旋转和比例不变性. SVD 基于特征值和特征向量进行变换,可以将非对称矩阵进行对角化,不仅限于方阵[14]. SVD 公式如(3)所示:

$$A=USV^T=\begin{bmatrix}u_{1,1}&\cdots&u_{1,n}\\u_{2,1}&\cdots&u_{2,n}\\\vdots&\vdots&\vdots\\u_{n,1}&\cdots&u_{n,n}\end{bmatrix}\begin{bmatrix}\sigma_1&0&\cdots&0\\0&\sigma_2&\cdots&0\\\vdots&\vdots&\ddots&\vdots\\0&0&\cdots&\sigma_n\end{bmatrix}\begin{bmatrix}v_{1,1}&\cdots&v_{1,n}\\v_{2,1}&\cdots&v_{2,n}\\\vdots&\vdots&\vdots\\v_{n,1}&\cdots&v_{n,n}\end{bmatrix} \tag{3}$$

式中:A 为 $n\times n$ 的矩阵,U,V 为正交特征向量矩阵,S 为奇异值矩阵. S 的奇异值作为样本特征来训练 LS-SVM.

2.4 最大水印嵌入强度

水印嵌入强度指水印嵌入过程中对载体图像的修改程度[15]. 水印嵌入强度越大,水印算法的鲁棒性越强,透明性越差. 本文将 SVD 的奇异值作为特征向量,与图像子块中心像素的最大水印嵌入强度组成样本来训练 LS-SVM 分类器,从而实现自适应的最大水印嵌入强度算法.

2.5 LS-SVM 分类器

LS-SVM 是支持向量机(SVM)的改进. LS-SVM 将 SVM 的二次规划问题变成了求解线性方程组的问题,降低了算法的复杂度,适用性更强[16]. 本文利用 LS-SVM 来求解最大水印嵌入强度和提取水印. 训练 LS-SVM 的特征向量为 SVD 的奇异值向量. 在求解自适应最大水印嵌入强度时,样本由特征向量和图像子块中心像素对应的最大水印嵌入强度组成;在提取水印时,样本由带水印载体图像和水印对应的样本标签值组成.

3 基于机器学习的数字水印算法

本文水印算法分为自适应最大水印嵌入强度、水印嵌入、LS-SVM 分类器的训练和水印提取四部分. 水印嵌入前先将其进行混沌加密,利用 LS-SVM 分类器得到各分块图像的自适应最大水印嵌入强度. 水印嵌入时,先对载体图像进行一级离散小波变换,选取 LL 子带图像将其进行分块处理,再对各分块进行奇异值分解后嵌入加密水印. 最后利用训练好的 LS-SVM 分类器对奇异值向量的标签进行估计来实现水印的盲提取. 算法的流程图如图 1 所示.

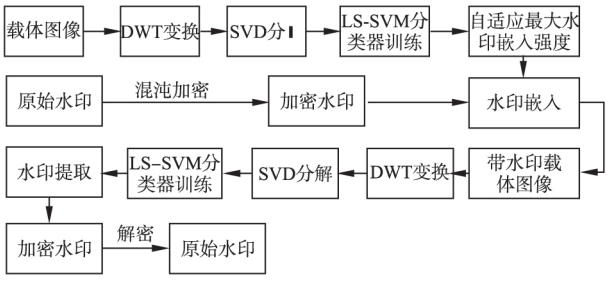


图 1 基于机器学习的数字水印算法流程图

Fig. 1 Flow chart of the digital watermarking algorithm based on machine learning

3.1 自适应最大水印嵌入强度

自适应最大水印嵌入强度算法根据不同图像子块的相关特征来计算对应的最大水印嵌入强度,利用 LS-SVM 分类器进行预测分类. 本文将像素灰度值归一化处理,故水印嵌入强度范围为[0,1],设定步长为 0.01 进行分析. 具体实现步骤如下:

Step 1. 对 64×64 载体图像 A 进行一级小波变换,选取 LL 子带对其进行 4×4 分块处理.

Step 2. 对图像子块 A_{ij} 进行奇异值分解,即 $A_{ij}=U_{ij}S_{ij}V_{ij}^T$.

Step 3. 将每个 S_{ij} 矩阵的对角线元素作为 A_{ij} 的特征向量 $V_{ij}=(\sigma_1,\sigma_2,\sigma_3,\sigma_4)$.

Step 4. 初始化水印嵌入强度 α 与结构相似度(SSIM)判断阈

值β,水印嵌入强度设定为 [0.3,1],α 初始化为 1,β 设定为 0.90.

Step 5. 按照式(4)的规则修改 A_{ij} 的中心像素值 $A_{ij}(4,4)$,A_{ij}^* 为修改后的图像子块.

$$A_{ij}^*(4,4) = A_{ij}(4,4) + \alpha \tag{4}$$

Step 6. 计算 A_{ij} 与 A_{ij}^* 的结构相似度 $\mathrm{SSIM}(A_{ij},A_{ij}^*)$,设定权重参数 $a=b=r=1$.

$$\mathrm{SSIM}(A_{ij},A_{ij}^*) = [l(A_{ij},A_{ij}^*)]^a \times [c(A_{ij},A_{ij}^*)]^b \times [s(A_{ij},A_{ij}^*)]^r \tag{5}$$

$$l(A_{ij},A_{ij}^*) = \frac{2\mu_{A_{ij}}\mu_{A_{ij}^*} + c_1}{\mu_{A_{ij}}^2 + \mu_{A_{ij}^*}^2 + c_1} \tag{6}$$

$$c(A_{ij},A_{ij}^*) = \frac{2\sigma_{A_{ij}}\sigma_{A_{ij}^*} + c_2}{\sigma_{A_{ij}}^2 + \sigma_{A_{ij}^*}^2 + c_2} \tag{7}$$

$$s(A_{ij},A_{ij}^*) = \frac{\sigma_{A_{ij}A_{ij}^*} + c_3}{\sigma_{A_{ij}}\sigma_{A_{ij}^*} + c_3} \tag{8}$$

式中,$\mu_{A_{ij}}$,$\mu_{A_{ij}^*}$ 分别为 A_{ij},A_{ij}^* 的均值,$\sigma_{A_{ij}}^2$,$\sigma_{A_{ij}^*}^2$ 分别为 A_{ij},A_{ij}^* 的方差,$\sigma_{A_{ij}A_{ij}^*}$ 为 A_{ij} 和 A_{ij}^* 的协方差,c_1,c_2,c_3 为常数,一般 $c_3 = c_2/2$.

Step 7. 若 $\mathrm{SSIM}(A_{ij},A_{ij}^*) < \beta$ 且 $\alpha > 0.3$ 时,更新 α 为 $\alpha - 0.01$,重新执行 Step5.

Step 8. 若 $\mathrm{SSIM}(A_{ij},A_{ij}^*) > \beta$ 或 $\alpha = 0.3$,则 α 为图像子块 A_{ij} 对应的最大水印嵌入强度值,记为 α_{ij}.

Step 9. 将特征向量 V_{ij} 与最大水印嵌入强度 α_{ij} 组成样本$\{V_{ij},\alpha_{ij}\}$,生成样本集合 A_n. 从中随机选取 80% 的样本生成训练样本集 A_{Train},剩余 20% 组成测试样本集 A_{Test}. 利用 A_{Train} 训练 LS-SVM,利用 A_{Test} 对训练好的模型进行测试. 输入特征向量可得最大水印嵌入强度.

3.2　水印的嵌入

选择 512×512 的图像作为载体,64×64 的图像作为水印. 首先将载体图像进行一级离散小波变换,选取 LL 子带将其进行 4×4 分块处理,然后将各分块进行奇异值分解,通过自适应最大水印嵌入强度算法计算各分块中心像素对应的最大嵌入强度,最后嵌入混沌加密的水印. 具体步骤如下:

Step 1. 对载体图像 I 进行一级小波变换,将 LL 子带进行 4×4 分块,并记为 LL_{ij},其中 $i \leqslant N$, $j \leqslant N$,N 为分块的个数.

Step 2. 对每一个 LL_{ij} 做 SVD 变换,即 $LL_{ij} = U_{ij}S_{ij}V_{ij}^T$.

Step 3. 选取 S_{ij} 的主对角线元素作为 LL_{ij} 的特征向量 $V_{ij} = (\sigma_1,\sigma_2,\sigma_3,\sigma_4)$.

Step 4. 根据自适应最大水印嵌入强度算法,输入特征向量 V_{ij} 得到图像子块中心像素的水印强度 α_{ij},按照按行优先原则,对特征向量的 σ_2 进行修改以嵌入水印信息,具体操作如公式(9)所示:

$$\begin{cases} \sigma_2 = \begin{cases} \mu + \alpha_{ij}, & \text{if } W_{ij} > 0 \text{ and } \sigma_2 < \mu \\ \mu - \alpha_{ij}, & \text{if } W_{ij} < 0 \text{ and } \sigma_2 > \mu \end{cases} \\ \mu = \dfrac{1}{4} \times \sum_{i=1}^{4} \sigma_i \end{cases} \tag{9}$$

Step 5. 由嵌入水印后的 V_{ij}' 推得 S_{ij}',进行 SVD 反变换 $U_{ij}S_{ij}'V_{ij}^T = LL_{ij}'$.

Step 6. 由 LL_{ij}' 组合可得 LL' 子带,与 HL、LH 和 HH 分量组合后经 DWT 逆变换可得嵌入水印后的载体图像 I'.

3.3　LS-SVM 分类器的训练

将未受攻击和经受不同攻击的带水印载体图像作为样本训练 LS-SVM,并将其分别记为 I',I_1',I_2',I_3',I_4',I_5',I_6',I_7'. 本文选取七类攻击方法: 高斯噪声(0.01)、椒盐噪声(0.01)、中值滤波(3×3)、JPEG 压缩(70)、逆时针旋转(5°)、剪切(左上 1/8)和缩放(0.5). 具体训练步骤如下:

Step 1. 对上述八类载体图像进行一级离散小波变换,分别选取 LL 子带并记为 LL',LL_1',LL_2',LL_3',LL_4',LL_5',LL_6',LL_7'.

Step 2. 分别将 LL',LL_1',LL_2',LL_3',LL_4',LL_5',LL_6',LL_7' 进行 4×4 分块处理,对每一分块做 SVD 变换 $LL_{ij} = U_{ij}S_{ij}V_{ij}^T$.

Step 3. 选取 S 分量中的对角线向量作为特征向量 $V_{ij} = (\sigma_1,\sigma_2,\sigma_3,\sigma_4)$ 共 8 组.

Step 4. 由水印图像 W_{ij} 映射得到样本集标签,如式(10)所示:

$$Tag_{ij} = \begin{cases} 1, & W_{ij} == 1 \\ 2, & W_{ij} == 0 \end{cases} \tag{10}$$

Step 5. 由上面产生的 8 组样本构成 LS-SVM 的训练样本集 $Train_{ij}^k = \{V_{ij},Tag_{ij}\}$,$k=1,2,3,4,5,6,7,8$,$i \leqslant N$,$j \leqslant N$.

Step 6. 由样本集 $Train_{ij}^k$ 训练可得 LS-SVM 分类器.

3.4　水印的提取

水印提取为水印嵌入的逆过程. 首先对嵌入水印后的载体图像进行一级小波变换,然后对 LL 子带按照 4×4 进行分块后进行 SVD 变换. 利用训练好的 LS-SVM 分类器提取水印. 具体步骤如下:

Step 1. 对嵌入水印后的载体图像 I' 进行一级离散小波变换,选取 LL 子带并得到 4×4 的分块 LL_{ij}',其中 $i \leqslant N$, $j \leqslant N$.

Step 2. 对 LL_{ij}' 进行 SVD 变换 $LL_{ij}' = U_{ij}'S_{ij}'V_{ij}'^T$,选取 S 的对角线向量作为特征向量 $V_{ij}' = (\sigma_1',\sigma_2',\sigma_3',\sigma_4')$.

Step 3. 利用训练好的 LS-SVM 对 V_{ij}' 的标签值进行估计,输出记为 Tag_{ij}'.

Step 4. 由标签值 Tag_{ij}' 提取水印图像 W_{ij}',如式(11)所示:

$$W_{ij}' = \begin{cases} 1, \text{if } Tag_{ij}' == 1 \\ 0, \text{if } Tag_{ij}' == 2 \end{cases} \tag{11}$$

4　实验分析

4.1　性能参数

4.1.1　透明性参数

选取原始载体图像和含水印载体图像的平均结构相似度(MSSIM)[17] 与峰值信噪比(PSNR)[18] 来评价水印算法的透明性. MSSIM 和 PSNR 均是越大代表水印透明性越好. PSNR 反映了图像数据变化的统计平均量,PSNR≥40dB 时人眼无法察觉图像是否有改动. MSSIM 与 PSNR 的计算公式分别如式(12)和式(13)所示:

$$\mathrm{MSSIM} = \frac{\sum_{i=1}^{N}\mathrm{SSIM}(I,I')}{N} \tag{12}$$

式中,I 为原始载体图像,I' 为含水印载体图像,N 为图像子块个数.

$$\mathrm{PSNR} = 10 \times \log_{10}\frac{255^2}{\mathrm{MSE}} \tag{13}$$

式中,MSE 为 I 与 I' 的平均绝对误差[18],公式如(14)所示:

$$\mathrm{MSE}=\frac{1}{L\times K}\sum_{i=1}^{L}\sum_{j=1}^{K}|I_{ij}-I'_{ij}|^2 \tag{14}$$

式中，L 和 K 分别代表载体图像的长和宽.

4.1.2 鲁棒性参数

鲁棒性反映水印算法的抗攻击能力，选取原始水印和提取水印的相似度系数(NC)[19]和比特错误率(BER)[20]对算法的鲁棒性进行评价. BER 反映了提取的水印信息与原始水印信息之间的差异，BER 越小，算法鲁棒性越强. NC 代表了原始水印与提取水印的相似度，NC 越大鲁棒性越好. BER 和 NC 的公式分别如式(15)和式(16)所示:

$$\mathrm{BER}=\frac{\sum_{i=1}^{W_L}\mathrm{abs}(\overrightarrow{W}-\overrightarrow{W'})}{W_L} \tag{15}$$

式中，W_L 为水印信息长度.

$$\mathrm{NC}=\frac{\sum_{i=1}^{n_L}\sum_{j=1}^{n_K}\frac{|W(i,j)+W'(i,j)|}{2}}{n_L\times n_K} \tag{16}$$

其中 W 为原始水印，W' 为提取水印，n_L、n_K 是水印图像长与宽.

4.2 实验仿真

实验基于 python3.6 在 4GB RAM、1.6GHz、内核 i5 处理器的 PC 机进行来评估本文算法的性能. 采用 3 张 512×512 的标准测试图像为载体图像，如图 2 所示; 采用 64×64 的二值图像为水印图像，如图4(a)所示. 图3为本文算法嵌入水印

(a) Lena (b) peppers (c) plane

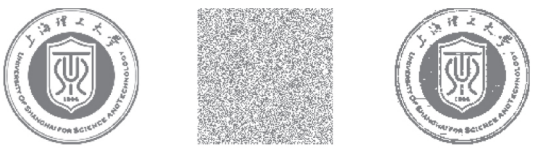
图 2 原始载体图像

Fig. 2 Original host images

(a) Lena (b) peppers (c) plane

图 3 带水印载体图像

Fig. 3 Host images with the watermark



(a) 原始水印 (b) 混沌加密水印 (c) 提取水印

图 4 不同阶段的水印图像

Fig. 4 Watermark images in different stages

后的载体图像，图 4(b)和图 4(c)分别为加密水印和无攻击时提取出的水印图像. 为了进一步评估本文算法性能，与其他 3 个算法进行了实验对比.

4.2.1 透明性测试

选取图 2 的载体图像用于透明性测试，在未受攻击的情况下，4 种算法的透明性参数 MSSIM 和 PSNR 如表 1 和表 2 所示. 图 5 为 MSSIM 和 PSNR 的对比图.

表 1 4 种算法的 MSSIM 对比

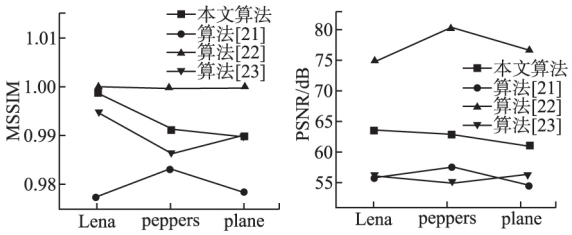
Table 1 Comparison of MSSIM among four algorithms

图像	本文算法	算法[21]	算法[22]	算法[23]
Lena	0.9990	0.9777	0.9998	0.9951
peppers	0.9912	0.9834	1.0000	0.9865
plane	0.9899	0.9786	0.9999	0.9903

表 2 4 种算法的 PSNR 对比

Table 2 Comparison of PSNR among four algorithms

图像	本文算法	算法[21]	算法[22]	算法[23]
Lena	63.71	55.88	74.93	56.22
peppers	62.99	57.78	80.48	55.24
plane	61.25	54.79	76.92	56.35



(a)MSSIM对比图 (b)PSNR对比图

图 5 透明性参数对比图

Fig. 5 Comparison of transparency parameters

4.2.2 鲁棒性测试

选取 Lena 图像用于鲁棒性测试，对图像添加攻击后提取水印，4 种算法的鲁棒性参数 BER 和 NC 如表 3 和表 4 所示.

表 3 Lena 图像的 BER 对比

Table 3 BER comparison of Lena image

攻击方法	本文算法	算法[21]	算法[22]	算法[23]
无攻击	0.0000	0.0009	0.0005	0.0000
高斯噪声 0.01	0.0098	0.0110	0.0298	0.0046
椒盐噪声 0.01	0.0086	0.0091	0.0053	0.0035
中值滤波 3×3	0.1015	0.0347	0.0528	0.0192
JPEG 压缩 70	0.0129	0.0096	0.0001	0.0261
逆时针旋转 5°	0.0071	0.0135	0.0001	0.0037
剪切(左上 1/8)	0.0046	0.0218	0.0134	0.0299
缩放 0.5	0.0020	0.0174	0.0257	0.0304

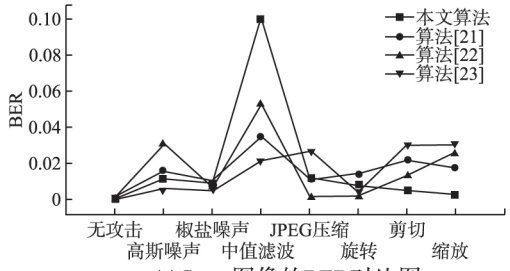
表 4 Lena 图像的 NC 对比

Table 4 NC comparison of Lena image

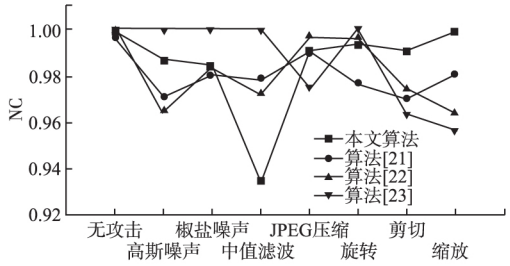
攻击方法	本文算法	算法[21]	算法[22]	算法[23]
无攻击	1.0000	0.9961	0.9993	1.0000
高斯噪声 0.01	0.9836	0.9755	0.9661	1.0000
椒盐噪声 0.01	0.9841	0.9802	0.9903	1.0000
中值滤波 3×3	0.9277	0.9783	0.9722	1.0000
JPEG 压缩 70	0.9888	0.9899	0.9964	0.9748
逆时针旋转 5°	0.9901	0.9764	0.9959	1.0000
剪切(左上 1/8)	0.9899	0.9698	0.9745	0.9633
缩放 0.5	0.9992	0.9799	0.9638	0.9565

图 6 给出了 BER 和 NC 的对比图. 本文选取 7 种攻击方法，分

别为高斯噪声 0.01、椒盐噪声 0.01、剪切(左上 1/8)、中值滤波 3×3、逆时针旋转 5°、JPEG 压缩 70 和缩放 0.5.



(a) Lena图像的BER对比图



(b) Lena图像的NC对比图

图 6　鲁棒性参数对比图

Fig. 6　Comparison of robustness parameters

4.2.3　算法复杂度测试

选取 Lena 图像进行算法复杂度测试,将 512×512Lena 图像嵌入和提取水印整个过程的运行时间作为判定标准. 4 个算法的运行时间对比如表 5 所示.

表 5　4 个算法的运行时间比较

Table 5　Comparison of running time among four algorithms

算法	本文算法	算法[21]	算法[22]	算法[23]
运行时间(s)	0.4010	0.7107	0.5924	0.3856

由表 1 和表 2 可知,本文算法具有很好的透明性,未受攻击时,透明性参数 MSSIM 大于 0.98,PSNR 达到 60dB 以上. 由表 3 和表 4 可知,本文算法也具有很高的鲁棒性. 在受到不同的攻击时,除了中值滤波,鲁棒性参数 BER 均小于 0.02,NC 均大于 0.95. 由表 5 可以看出,本文的算法复杂度也占有一定的优势,作为机器学习算法仍然保持较高的运行效率. 本文的自适应最大水印嵌入强度算法,在保证水印算法透明性的前提下尽可能增加水印的嵌入强度,提高了算法的鲁棒性.

与其他 3 个算法相比,本文算法分别表现出了一定的优势. 由图 5 可以看出,本文算法的透明性明显优于文献[21]和文献[23]的算法. 由图 6 可知,除了中值滤波,本文算法抵抗常规攻击的能力较高且保持均衡,适用性较广. 在面对剪切、缩放等几何攻击时,本文算法鲁棒性优于其他 3 个算法. 在算法复杂度方面,本文算法优于文献[21]和[22]的算法. 文献[22]虽然透明性优于本文算法,但本文算法的透明性早已超过 PSNR=40dB 的人眼可辨识范围,而且文献[22]的算法面对高斯噪声时鲁棒性不是很强. 另外,本文的自适应水印嵌入强度均大于其他 3 个算法,这是本文的一大优势. 综合来看,本文算法兼顾透明性、鲁棒性、嵌入强度以及时间复杂度,性能最佳.

本文算法也有不足之处,如抵抗中值滤波攻击时鲁棒性较弱,有待于提高.

5　结论与展望

本文提出了一种利用机器学习方法的图像数字水印算法,算法包括 LS-SVM 分类器的训练以及水印的嵌入和提取. 算法将 DWT 和 SVD 相结合嵌入水印,将 LS-SVM 分类器用于自适应最大水印嵌入强度算法和水印的盲提取. 最大水印嵌入强度算法使得水印算法保证透明性的前提下尽可能地提高鲁棒性. 实验证明,在水印嵌入强度范围为[0.3,1]的条件下,本文算法的透明性实现了最佳 PSNR=63.71dB 的效果,针对 JPEG 压缩、高斯噪声,尤其是旋转、剪切和缩放等常规攻击手段,算法能保持较强的鲁棒性.

本文算法针对中值滤波攻击时鲁棒性有待于提高,在未来的研究中,可针对中值滤波攻击进行算法改进,通过不断优化 LS-SVM 分类器参数来获得更高的透明性与鲁棒性. 今后的研究也可以考虑尝试其他机器学习算法与传统数字水印算法的结合,使其性能更佳,适用范围更广.

References:

[1] Sharma B, Dave M. Robust hybrid image and audio watermarking using cyclic codes and arnold transform [C]//2019 International Conference on Communication and Electronics Systems(ICCES), Coimbatore, India, 2019: 309-315.

[2] Patel F, Paunwala C, Vora A. A modified block based biometric fragile watermark technique [C]//2019 International Conference on Communication and Electronics Systems(ICCES), Coimbatore, India, 2019: 942-946.

[3] Ghaemi A, Danyali H. CRT-based robust data hiding method by extracting features in DCT domain [C]//2019 16th International ISC(Iranian Society of Cryptology) Conference on Information Security and Cryptology(ISCISC), Mashhad, Iran, 2019: 134-138.

[4] Gunjal B L, Mali S N. Comparative performance analysis of digital image watermarking scheme in DWT and DWT-FWHT-SVD domains [C]//2014 Annual IEEE India Conference(INDICON), Pune, 2014: 1-6.

[5] Liu S, Pan Z, Song H. Digital image watermarking method based on DCT and fractal encoding [J]. IET Image Processing, 2017, 11(10): 815-821.

[6] Guo Y, Li B. Blind image watermarking method based on linear canonical wavelet transform and QR decomposition [J]. IET Image Processing, 2016, 10(10): 773-786.

[7] Chen Qing, Guo Gong-xun. A wavelet coefficient adjustment watermarking algorithm based on modulus operation [J]. Journal of Chinese Computer Systems, 2019, 40(2): 157-162.

[8] Makbol N M, Khoo B E, Rassem T H. Block-based discrete wavelet transform-singular value decomposition image watermarking scheme using human visual system characteristics [J]. IET Image Processing, 2016, 10(1): 34-52.

[9] Aparna P P, Kishore V V. Biometric-based efficient medical image watermarking in E-healthcare application [J]. IET Image Processing, 2019, 13(3): 421-428.

[10] Zheng Pan-pan. Research on image digital watermarking based on

intelligent optimization and pattern recognition [D]. Xi'an: Northwest University, 2013.

[11] Islam M, Mallikharjunudu G, Parmar A S, et al. SVM regression based robust image watermarking technique in joint DWT-DCT domain [C]//International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), Kannur, 2017: 1426-1433.

[12] Rachmawanto E H, De Rosal I M S, Sari C A, et al. Block-based Arnold chaotic map for image encryption [C]//International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, Indonesia, 2019: 174-178.

[13] Run R S, Horng S J, Lai J L, et al. An improved SVD-based watermarking technique for copyright protection [J]. Expert Systems with Applications, 2012, 39(1): 673-689.

[14] Reddy T S, Reddy D S, Nihar A N, et al. Comparitive analysis on transformation based watermarking [C]//2nd International Conference on Signal Processing and Communication (ICSPC), Coimbatore, India, 2019: 356-360.

[15] Liu Yi-nan. Research on image digital watermarking algorithm based on machine learning [D]. Beijing: Beijing University of Posts and Telecommunications, 2014.

[16] Ji R, Wu Y, Qi K. Image inpainting with LS-SVM based on gradient information [C]//Chinese Automation Congress (CAC), Xian, China, 2018: 4050-4055.

[17] Hu H T, Hsu L Y, Chou H H. An improved SVD-based blind color image watermarking algorithm with mixed modulation incorporated [J]. Information Sciences, 2020, 519: 161-182.

[18] Alvin Wicaksana A, Prasetiyowati M I. Digital watermarking for color image using DHWT and LSB [C]//2019 5th International Conference on New Media Studies (CONMEDIA), Bali, Indonesia, 2019: 94-99.

[19] Su Q. Novel blind colour image watermarking technique using Hessenberg decomposition [J]. IET Image Processing, 2016, 10(11): 817-829.

[20] Ahmadi M, Norouzi A, Karimi N, et al. ReDMark: framework for residual diffusion watermarking based on deep networks [J]. Expert Systems with Applications, 2020, 146: 105-120.

[21] Vairaprakash S, Shenbagavalli A. A discrete rajan transform-based robustness improvement encrypted watermark scheme backed by support vector machine [J]. Computers & Electrical Engineering, 2017, doi: S0045790617331580.

[22] Reza S. H., Mohammad S. Transform-based watermarking algorithm maintaining perceptual transparency [J]. IET Image Processing, 2018, 12(5): 751-759.

[23] Xu Wei, Liu Ying, Zhu Ting-ge. Image hash watermarking algorithm based on DCT and SVD [J]. Computer Engineering and Design, 2020, 41(1): 145-149.

附中文参考文献:

[7] 陈 青,郭功勋. 一种基于模运算的小波系数调整水印算法 [J]. 小型微型计算机系统, 2019, 40(2): 157-162.

[10] 郑盼盼. 基于智能优化和模式识别的图像数字水印研究 [D]. 西安: 西北大学, 2013.

[15] 刘一楠. 基于机器学习的图像数字水印算法研究 [D]. 北京: 北京邮电大学, 2014.

[23] 徐 伟,刘 颖,朱婷鸽. 基于 DCT 和 SVD 的图像哈希水印算法 [J]. 计算机工程与设计, 2020, 41(1): 145-149.