Scientific and Social Research, 2023, Volume 5, Issue 12
https://ojs.bbwpublisher.com/index.php/SSR

Digital Image Copyright Protection System Through Blockchain and Digital Watermarking

Feng Liu*, Xinyu Pan, Baowei Cheng, Haoxin Wang, Chaoyi Deng

School of Cyberspace Security, Zhengzhou University, Zhengzhou 450000, Henan Province, China

Corresponding author: Feng Liu, 15737132818@163.com

Abstract: In this paper, we propose a digital image authentication model with an alliance chain through blockchain + InterPlanetary File System (IPFS) technology and ResNet-50 convolutional neural network digital watermarking technology. The blockchain + distributed storage method is used to solve the problem of large-scale data uplink, and the original data is stored in the IPFS distributed system. After the image is uploaded, the image digital works are embedded through the deep-learning model based on ResNet-50 to form a new carrier digital image embedded with a watermark. When infringement is found, watermark extraction is performed on the infringing image to obtain the original watermark. By analyzing the information recorded in the blockchain and tracing the infringement on the blockchain, the copyright of users' digital images can be protected.

Keywords: Blockchain; Artificial intelligence; Digital watermarking

Online publication: December 27, 2023

1. Introduction

In recent years, there have been more and more cases of intellectual property infringement. Statistics show that Chinese courts received 526165 new cases of various types of intellectual property rights in 2022, and the Internet has become one of the most important places where violations of intellectual property rights infringement occur.

Xu et al. [1] proposed a method and system for digital copyright protection and combined auction based on blockchain. The approach involves decentralizing the management of copyright data rights through the workload-proof mechanism of blockchain. The broadcast between blockchain nodes is used to realize the decentralization of copyright data storage.

Li [2] proposed a digital watermarking algorithm based on carrier image preprocessing and DCT-SVD transform. The algorithm exhibited robustness against various attacks, including noise attacks, mean filtering attacks, clipping attacks, and rotation attacks, producing a clearer watermark image even after such attacks. The Normalized Cross-Correlation (NC) value of the extracted watermark after different attacks was greater than 0.86.

Based on the existing technology, we used the pre-trained classification neural network ResNet-50 to extract image features. This process was combined with fabric and the InterPlanetary File System (IPFS) to facilitate large-scale data uplink. The evaluation of node behavior was conducted through a credit election approach. Additionally, the watermark embedding technology was enhanced using gradient descent on the updated image, addressing the inefficiency, high cost, and elevated risk associated with traditional centralized copyright management organizations. This solution aimed to mitigate challenges in digital image copyright protection, as illustrated in **Figure 1**.

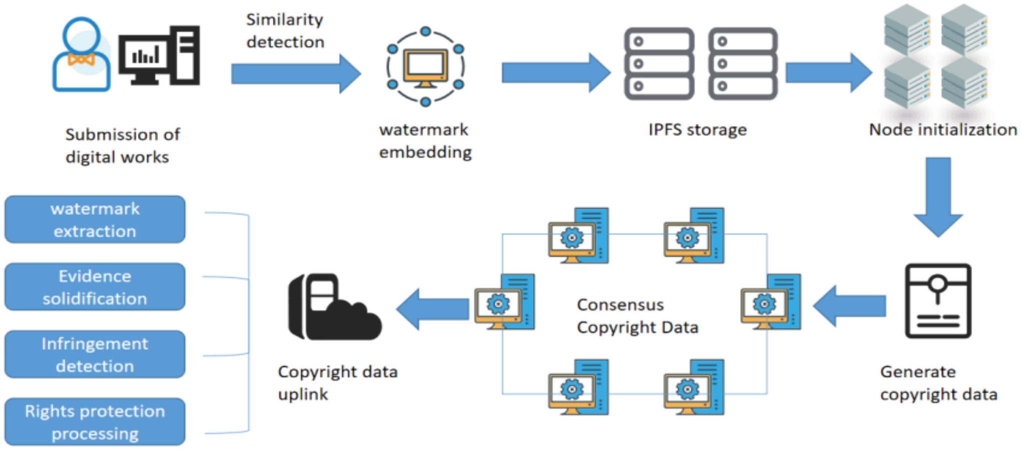


Figure 1. Digital image upload and storage flow chart

2. Method

2.1. Digital watermarking technology based on ResNet-50

A pre-trained ResNet-50 classification neural network was used as the image feature extraction model. It utilizes the output features from the final convolutional layer of ResNet-50 [3]. To ensure uniformity in the distribution of features, a normalization layer was added, employing PCA whitening transform technology. Then, the watermark was transformed into a random orthogonal carrier vector using an algorithm, and it was embedded into the feature space of the image. In the embedding process, a mark with image optimization and data enhancement was designed to optimize the watermark and improve its robustness. ResNet50 is used to extract the features of the watermark and perform inverse transformation according to the orthogonal carrier vector, as shown in **Figure 2**.

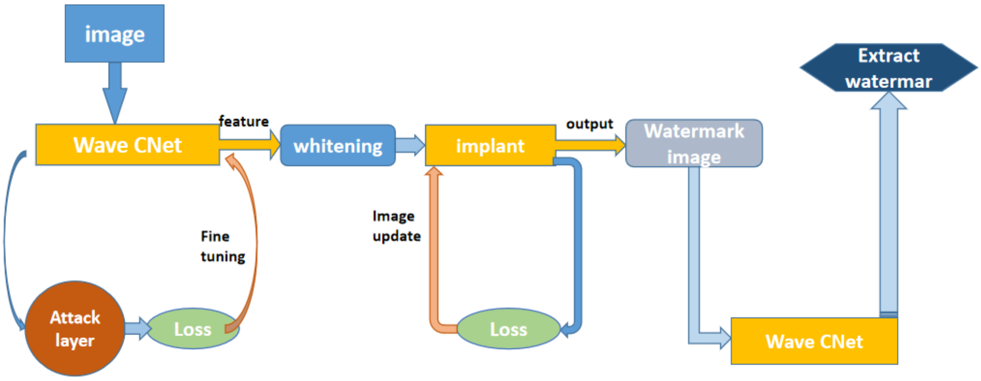


Figure 2. Structure of the digital watermarking model

2.1.1. Whitening transformation

The output data of the last ordinary convolutional layer of the ResNet50 neural network had 2048 features. The data was first centralized by subtracting the mean on each dimension, removing the translation effect to position the data’s center at the origin.

Next, the covariance matrix of the data was calculated, and this matrix was decomposed into eigenvalues and eigenvectors. By sorting the eigenvectors according to the eigenvalues, a set of principal components is obtained [4].

In PCA whitening, the feature vector corresponding to the first *k* largest eigenvalues was chosen, typically retaining the feature vector that preserves most of the information in the original data. *U_reduce* is the first *k* column of the eigenvector matrix, while *X* represents the original data matrix and *X_white* is the dimensionality-reduced data matrix. Finally, the data was normalized so that the variance of each feature was 1 and the correlation between the different features was 0. *D* was the diagonal matrix of the eigenvalue matrix. Taking the square root of the reciprocal of *D*, a whitening matrix with equal variance was obtained in each dimension.

$$\text{X\_white}=D^{-\frac{1}{2}} * U\_reduce^{T*X}$$

2.1.2. Transfer Learning

Watermark images are often subjected to malicious attacks. Several examples of attacks are shown in **Figure 3** and **Table 1**. Hence, we designed and added an attack simulation layer to perform different types of attacks on the watermarked image in each minimum batch of training [5].

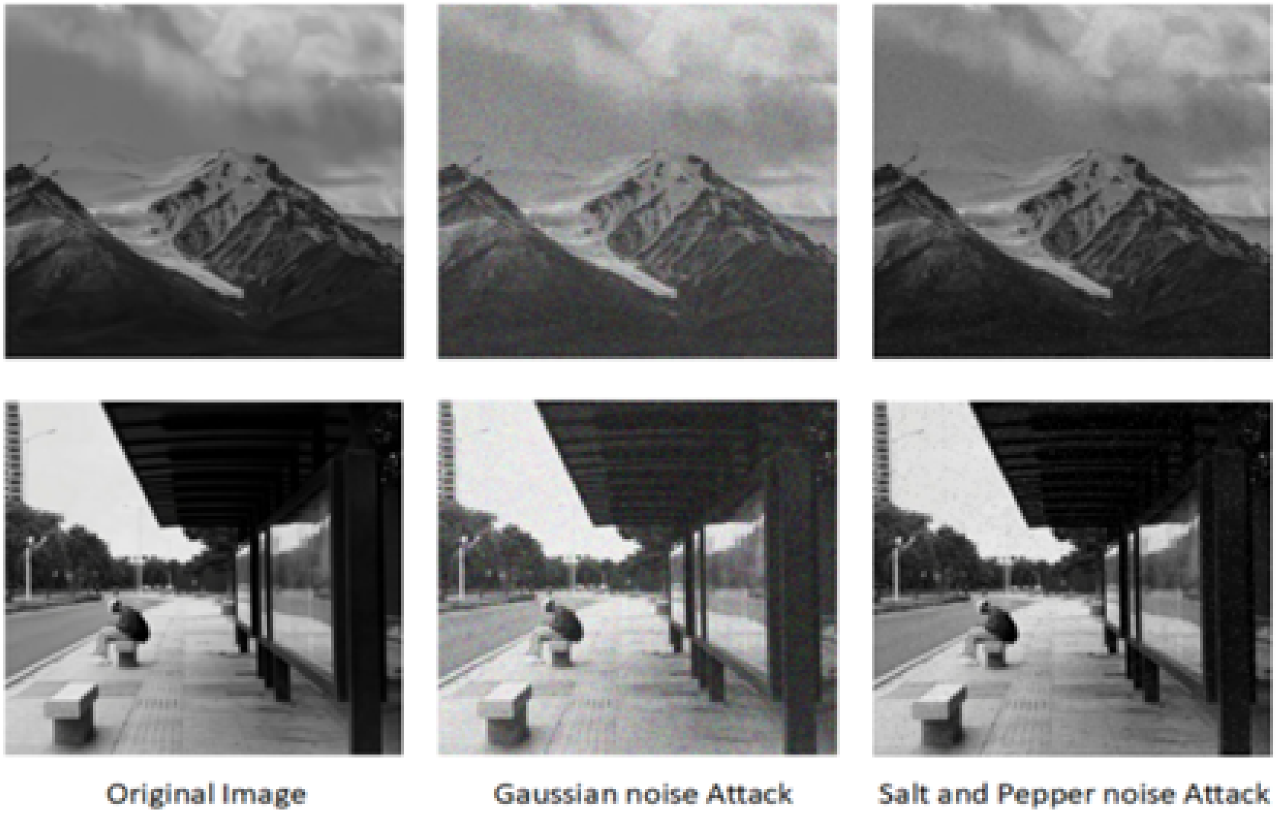


Figure 3. Examples of attacks

Table 1. Types of attacks on watermarked images

Attack	Related parameters	Proportion
None	-	1/3
Salt and pepper noise attack	p = 0.1	1/6
Gaussian noise	σ = 0.2	1/6
JPEG compression	Quality = 50	1/6
Dropout	p = 0.3	1/6

Two loss functions were set up to fine-tune the network. The similarity between the watermark image and the original image is measured by the MSE loss function shown in Equation (1), where M and N are the resolution of the original image, I_h is the original image, and I_{WI} is the processed watermark.

In addition, the MAE function was used as a loss function between the extracted watermark and the input watermark displayed in Equation (2), where X is the size of the watermark, WM_0 is the original watermark, and WM_E is the extracted watermark.

$$L_1 = \frac{1}{\mathrm{MN}} \sum_{i,j}^{\mathrm{MN}} [I_h(\mathrm{i,j})\text{-}\mathrm{I_{WI}}(\mathrm{i,j})]^2 \qquad (1)$$

$$L_2 = \frac{1}{X} \sum_{i}^{X} |\mathrm{WM}_O(i)\text{-}\mathrm{WM}_E(i)| \qquad (2)$$

$$L_3 = \lambda_1 L_1 + \lambda_2 L_2 \qquad (3)$$

The combination of the above functions (L_1 and L_2) determines the loss function of the entire network, where the coefficients (λ_1 and λ_2) in Equation (3) are super-parameters to adjust invisibility and robustness.

In this project, the last full connection layer and pooling layer of ResNet-50 neural network were removed, the last convolution layer was connected with the normalization layer and attack layer described above, and the coco dataset was used for training.

2.1.3. Watermark marking

In order to improve the quality of watermark, two loss functions were defined to measure the effect of the watermark. $L_W :F \rightarrow D$ indicates how far the feature x ∈ f to be captured is from region D and calculates the error during watermark embedding and extraction. Loss function, $L_i :I \times I \rightarrow R_+$, was used to measure the difference between the predicted value and the actual value.

In order to improve the robustness of the watermark image, we also defined a set of attacks T, including rotation, cropping, blur, etc. $T_r(I,t) \in I$ means that transform $t \in T$ is applied to image I. The watermark uses data enhancement. Losses L_W and L_i were combined as $L(I,I_0,t): = \lambda L_W(\varphi(T_r(I,t))) + L_i(I,I_0)$.

Among them, L_W aims to push any converted feature of I_W toward D, while L_i tends to be low distortion. Equation (4) is the training method of the counterattack training.

$$I_W = ar\ gmin_{I \in C(I_0)} E_{t \sim T}[L(I,I_0,t)] \qquad (4)$$

where $C(I_0) \subset I$ is the set of received images, which is defined by the difference of the two-step normalization applied to the pixel direction ($\delta = I - I_0$). SSIM was used in the training process, which scales δ in the pixel direction to hide the information in the perceptually invisible area of the image. A minimum target

PSNR was set to control the image quality. If the target was exceeded, the function was used to rescale δ.

2.1.4. Watermark extraction

In the process of watermark extraction, we used the ResNet-50 neural network model to extract the input image features embedded with the watermark and form the feature vector. During decoding, the carrier matrix was used to match the feature vector of the embedded watermark image with the message bits. Assuming that the message to be hidden was $m = (m_1, \ldots, m_k \in \{-1, 1\})^k$, the decoder retrieves $m = D(I)$. In this scenario, the key is represented by randomly sampled orthogonal vectors $a_1, \ldots, a_k \in R^d$. The encoding process involves projecting m onto a feature $\varphi(1)$ with the decoder given by: $D(I) = [sign\varphi(1)^T a_1) \ldots, sign\varphi(1)^T a_k)]$.

2.2. Large-scale data uplink based on fabric + IPFS

Blockchain + distributed storage is used to solve the problem of large-scale data chaining. The original data is stored within the IPFS distributed system, and the blockchain stores the address of the source file for permanent storage. Users can access this data anytime using the file's address information on the blockchain. Additionally, the file's fingerprint is stored on the blockchain, enabling users to verify the data on the chain and ensure its integrity and reliability.

2.2.1. Node initialization

Node initialization forms the foundation of model operation. Ordinary nodes, characterized by high credit values calculated by the OPBFT algorithm, transition into consensus nodes, while those with low credit values become non-consensus nodes. Diverging from the conventional Client/Server (C/S) architecture, this model operates on a Peer-to-Peer (P2P) network [6]. Both consensus nodes and non-consensus nodes are peer nodes that can communicate with each other.

Upon joining the network, the consensus node transmits version information indicating the maximum block height to other nodes for connection establishment. Connecting nodes compare the maximum height version information, enumerate blocks below the maximum height, and dispatch them to the newly joined consensus node. The node downloads any missing blocks to its local blockchain, ensuring consistency with the data status of other nodes.

2.2.2. Generating copyright data

Utilizing smart contracts and blockchain technology, the Generate Copyright (GC) function can autonomously generate copyright data. To fulfill its purpose, copyright data must be deployed on the blockchain, as illustrated in Figure 4.

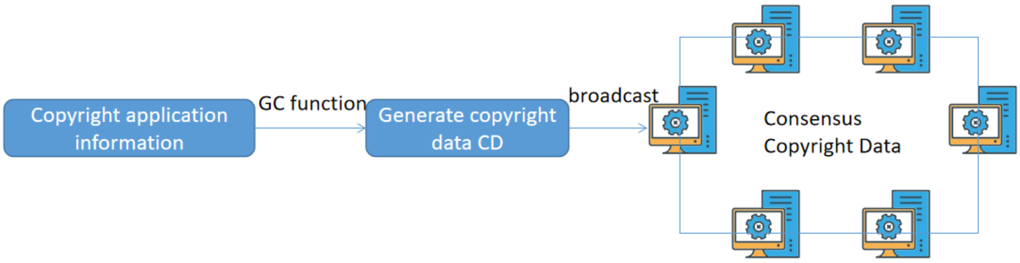


Figure 4. Generation of copyright data using a smart contract

The digital image works uploaded by users are output as copyright data (CD) by calling the add_Digital_to_Store function in GC and then broadcast the CD by calling the broadcast function. Upon receiving the CD, the consensus node will verify the digital signature's validity on the CD. If the validity check is successful, the node will package the CD into blocks, timestamp them, and subject them to the OPBFT algorithm for consensus.

2.2.3. Consensus copyright data

PBFT algorithm can accept Byzantine nodes, which account for one-third of the total number of system nodes at most [7]. In other words, if more than two-thirds of the nodes are healthy, the whole system runs well. The consensus node set A receives the copyright data CD and checks the validity of the CD. After the CD is verified, the main node packs the CD into blocks and broadcasts them to the network. After receiving the block, the replica node checks whether the hash value of the previous block, the root hash value of the Merkel tree, and the block ID are consistent with the local block. If the results align, the block is accepted, and a commit message is broadcast. When more than two-thirds of the nodes in set A receive the commit message, it confirms a successful CD consensus. Nodes involved in the consensus then incorporate blocks containing CD into the blockchain.

2.2.4. Copyright data uplinking

The blockchain uses the Merkle tree to store CD. The weight confirmation model network calculates the hash value (h_i) of CD, then combines the two calculated hash values to calculate the hash value (h'_i) again according to the formula, and calculates the root node hash value h_{root} of Merkle tree through layer-by-layer hash calculation.

$$h_i = H(RD_{\mathrm{i}}) \quad (5) \qquad h'_i = H(h_{\mathrm{i}} + h_{\mathrm{i+1}}) \quad (6)$$

In the Merkle tree, when a node tampers with a CD with h_{root}, the corresponding hash value will change, so the Merkle tree can prevent the copyright data on the blockchain from being tampered with. *root* is added to the block header as the root value of the Merkle tree. A block is composed of a block header and a block body. Each block is connected with the hash value field of the previous block to form a blockchain. The timestamp provides reliable and unalterable evidence of the registration time of digital image copyright information.

3. Experiment

In this project, the open-source and free Hyperledger Fabric framework served as the underlying blockchain platform. The Go language was utilized to write smart contracts for manipulating copyright information on the chain. The MySQL database stored user and administrator account information, and the value of the hash function after image upload (to facilitate similarity detection). The backend incorporated a watermark function using the PyTorch library in Python to construct an artificial intelligence model for watermark embedding. The front end was developed using HTML5, CSS3, JavaScript, and the Django framework, offering users an aesthetically pleasing and user-friendly interface. The detailed architecture of the digital image copyright protection system is depicted in **Figure 5**.

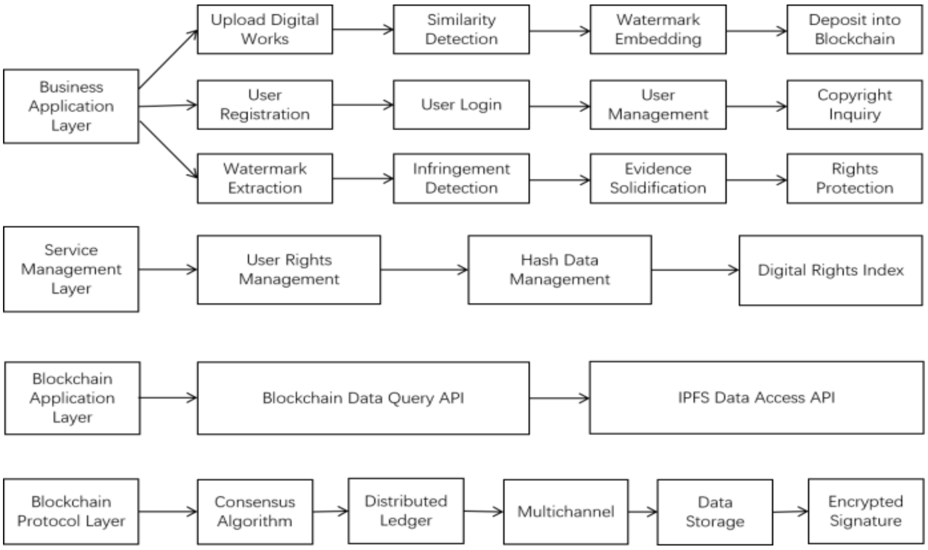


Figure 5. Digital image copyright protection system architecture

We built the whole system on Tencent Cloud, Alibaba Cloud, and other multi-cloud service platform servers. each node was equipped with a basic configuration of a 1-core CPU, 1GB RAM, 40GB SSD, 100Mbps peak bandwidth, and a 500GB traffic package. The front-end server was established using a local virtual machine with a configuration of 4 cores and 4GB of running memory.

3.1. Blockchain algorithm verification

In Experiment 1, four virtual machines and eight virtual machines were used to simulate network nodes for testing. The client node initiates multiple transactions. After receiving the message, the master node broadcasts the message to the whole network. After receiving the message, the replica node first verifies the signature of the message. If it passes the verification, it creates a transaction block and sends the message to other consensus nodes.

The total number of transactions generated was counted at intervals of 5 seconds, 15 seconds, 25 seconds, 35 seconds, 45 seconds, and 55 seconds respectively. In Experiment 2, multiple groups of nodes were used to test the consensus speed of transactions. Four machines are used to compare the traditional PBFT algorithm and OPBFT algorithm. The selected throughput comparison chart is shown in **Figure 6**.

From the throughput comparison chart of the four machines, it is evident that the throughput of the OPBFT algorithm was much higher than that of the traditional PBFT algorithm.

The consensus speed of transactions with different numbers of nodes is shown in **Figure 7**.

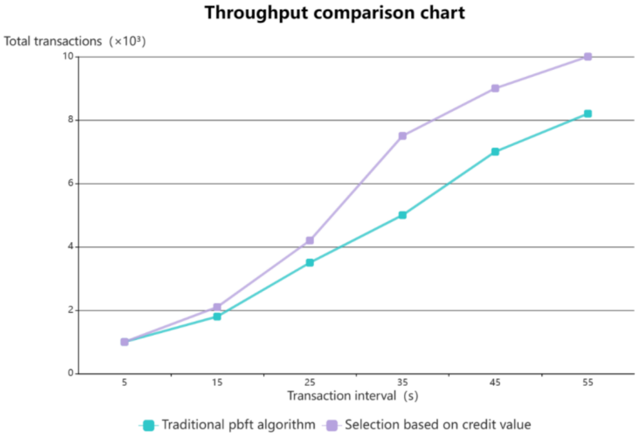


Figure 6. Throughput comparison chart

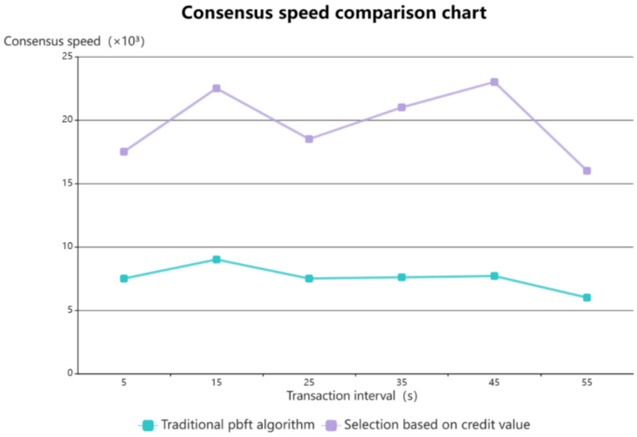


Figure 7. Consensus speed comparison chart

3.2. Evaluation index of digital watermark

300 pictures in the COCO dataset were selected, and attack processing was conducted on the model, such as blur, brightness, contrast, clipping, JPEG, scaling, rotation, etc., and the test results were obtained, as shown in **Figure 8** and **Figure 9**.

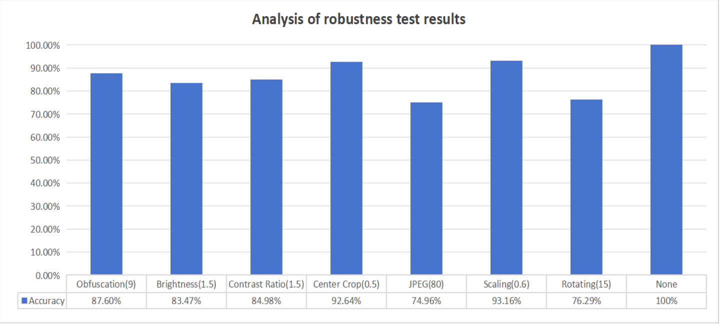
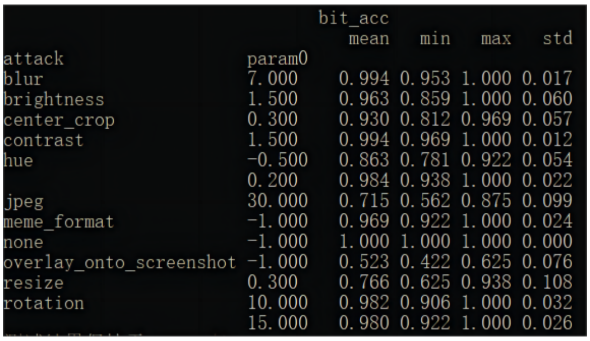


Figure 8. Overall display of test results



```
                                   bit_acc
                                      mean   min   max   std
attack                      param0
blur                        7.000    0.994 0.953 1.000 0.017
brightness                  1.500    0.963 0.859 1.000 0.060
center_crop                 0.300    0.930 0.812 0.969 0.057
contrast                    1.500    0.994 0.969 1.000 0.012
hue                         -0.500   0.863 0.781 0.922 0.054
                            0.200    0.984 0.938 1.000 0.022
jpeg                        30.000   0.715 0.562 0.875 0.099
meme_format                 -1.000   0.969 0.922 1.000 0.024
none                        -1.000   1.000 1.000 1.000 0.000
overlay_onto_screenshot     -1.000   0.523 0.422 0.625 0.076
resize                      0.300    0.766 0.625 0.938 0.108
rotation                    10.000   0.982 0.906 1.000 0.032
                            15.000   0.980 0.922 1.000 0.026
```

Figure 9. Analysis of robustness test results

Figure 8 and **Figure 9** show that, after undergoing most attack processing, the model achieved a recognition bit accuracy rate exceeding 80%, with the bit accuracy rate soaring to 96% after scaling processing. This surpassed classic models like Hidden. However, due to the quantitative embedding method used in the model, its robustness to rotation and JPEG compression was relatively weak, resulting in a bit accuracy of approximately 75%.

To ensure high image quality, the PSNR of the project was fixed at 40 dB. While theoretically capable of accommodating hundreds or thousands of bytes, incorporating higher bytes led to a decline in model robustness. Through testing and evaluation, it was found that the project performed exceptionally well with a 64-byte watermark. The project ran on an RTX 3050 video card with 4GB video memory. Partial images from the COCO dataset were chosen as the original images, and the embedding time was approximately 4.5 seconds per image, while the extraction time ranged in milliseconds.

4. Conclusion

The digital copyright industry has experienced rapid growth in the past decade, emerging as a crucial component of China's digital economy and a key driver of its development. With advancements in 5G, mobile Internet, and other information technologies, the digital content industry is currently in a phase of prosperity and expansion. However, this growth brings challenges such as copyright infringement and illegal copying. To address these issues, the project integrates blockchain + IPFS technology with deep learning, digital watermarking, and other technologies, aiming to enhance digital image copyright protection for users. In essence, this project is an innovative, practical, and forward-looking platform poised to fulfill the pressing need for copyright protection in the digital economy era. It not only offers enhanced protection and innovation incentives for digital creators but also provides crucial support and motivation for the growth of the digital culture industry, contributing robust technical safeguards for scientific and technological innovation.

Disclosure statement

The authors declare no conflict of interest.

References

[1] Xu Y, Qian Q, 2022, Blockchain-based Data Copyright Protection and Combinatorial Bidding. Journal of Shanghai University (Natural Science Edition), 28(3): 413–426.

[2] Li L, 2018, Research and Application of Blockchain Technology in Digital Copyright, dissertation, Northern University of Technology.

[3] Liu Y, Zhang J, Wu S, et al., 2021, Research on Digital Copyright Protection Based on the Hyperledger Fabric Blockchain Network Technology. PeerJ Comput Sci, 7: e709. https://www.doi.org/10.7717/peerj-cs.709

[4] Xu ZJ, Wang ZZ, Lu Q, 2011, Research on Image Watermarking Algorithm Based on DCT. Procedia Environmental Sciences, 10: 1129–1135. Proceedings of the 2011 3rd International Conference on Environmental Science and Information Application Technology ESIAT 2011.

[5] Pradhan C, 2020, Robust and Blind Watermarking Using Arnold 4D Cat Map in Discrete Wavelet. International Journal of Information Technology, 12(2): 593–597.

[6] Yorozu T, Hirano M, Oka K, et al., 1987, Electron Spectroscopy Studies on Magneto-Optical Media and Plastic Substrate Interface, Electron Spectroscopy Studies on Magneto-Optical Media and Plastic Substrate Interface, 2(8): 740–741.

[7] Young M, 1989, The Technical Writer’s Handbook. Mill Valley, University Science, California.

Publisher’s note

Bio-Byword Scientific Publishing remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.