Robust and High Capacity Watermarking for Image Based on DWT-SVD and CNN

Wenbo Zheng[‡*], Shaocong Mo[§*], Xin Jin*[¶†], Yili Qu*[††],Fei Deng[x], Jia Shuai[x], Zefeng Xie[x], Chengfeng Zheng[x] and Sijie Long [x]
[‡]School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, China
[§]College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China
[¶]School of Management, Huazhong University of Science and Technology, Wuhan 430074, China
[††]School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China
[x]School of Computer Science and Technology, Wuhan University of Technology, Wuhan 430070, China
*E-mail: jinxin_hust@163.com, quwillpower@gmail.com
*Wenbo Zheng and Shaocong Mo contribute equally to this study.
†Xin Jin and Yili Qu are both the corresponding author.

***Abstract*—Digital watermarking technology is of great importance to protect the copyright of the owners and authenticate the security of the media. As digital images are vulnerable to some common attacks during transmission, it is very necessary to design a watermarking algorithm which can resist all kinds of common attacks. Nowadays, most of the watermarking algorithms published rely much on the locations of the pixels for watermark embedding, which results in less robustness. And some algorithms took some measures to increase the ability to resist attacks, but the measures taken limited the algorithm in watermark embedding capacity. In this paper, a robust watermarking algorithm based on convolution neural network (CNN) is proposed. We introduce discrete wavelet transform (DWT) technology and singular value decomposition (SVD) technology, to achieve the embedding process of watermark. The network is established in the spatial domain based on the pixels' relationships of watermark, host image and watermarked image. After that, The pixels of the watermarked image are lightly modified with the network. In addition, some attacks are taken to the watermarked image. Simulation shows that proposed algorithm has good performance, the watermark extracted can be clearly identified.**

***Keywords*—Digital Image Watermarking, Convolutional Neural Networks, Discrete Wavelet Transform, Singular Value Decomposition**

I. Introduction

With the rapid development of computer Internet, digital watermarking technology is considered as an important branch of information hiding technology research, quickly become a hot field of multimedia information security[1], [2].

Digital watermarking technology is to embed a number of logo information (watermark) directly into the multimedia content. After embedding, the use value of original contents can not be affected and logo information can not be perceived by the perception system, only through a dedicated detector or reading device, watermark can be extracted[3], [4]. Unlike encryption, digital watermarking technology can't prevent the occurrences of piracy, but can determine whether the object is protected, monitor the spread of protected data, authenticate and illegally copy, resolve copyright disputes problems and provide evidence for the court[5], [6], [7].

Artificial neural network once was an important method in machine learning, which is a distributed information processing model that simulates the way of human brain work. In 2006, Professor Geoffrey Hinton [8] put forward the concepts of deep learning and improved the methods for training models, which broke the bottleneck of the development of traditional BP neural network. since then, deep learning has been becoming the focus of research. Convolution neural network (CNN) is a kind of deep neural network, which has become one of the hotspots in many fields of science[8].

In this paper, we try to combine digital watermarking technology with convolution neural network to improve the robustness of watermarking on the basis of existing watermarking algorithms.

II. Related Work

A. Singular value decomposition

SVD is one of the techniques for applications in image processing. Specifically, SVD has been used for image compression, image hiding and digital watermarking. For example, an image is denoted by a matrix A. SVD is defined: $A = U * S * V^T$. The image is decomposed into three matrices: two orthogonal matrices U, V and a diagonal matrix S. U, V are called left and right singular vectors. Coefficients diagonal matrix S are called the Singular Values (SVs) of the matrix A. In digital watermarking, SVD has some advantages: Reducing the size of signal embed in the image. And the SVs of the watermarked image are less influenced by attacks. It can be used as a robust feature in digital watermarking[9].

B. DWT-SVD

Ref[10], [11], [12] propose a blind digital image watermarking technique by combining DWT with SVD to improve the robustness and the capacity. In detail, SVs of watermarks are replaced with the suitable SVs of HH sub-bands of the original images. Additionally, our method generates keys that ensure the security for the watermarks in the embedding and the extraction process. Experiments on images for digital

978-1-5386-3758-6/18/$31.00

watermarking attacked by Stirmark Benchmark 4.0 tool, but it isn't show that the method is higher capacity as the same size as the water-making image.

C. Convolutional neural networks

Convolutional neural networks have recently achieved many successes in visual recognition tasks, including image classification, object detection, and scene parsing[13]. Ref[14] introduces a blind watermarking based on a convolutional neural network. We propose an iterative learning framework to secure robustness of watermarking. One loop of learning process consists of the following three stages: Watermark embedding, attack simulation, and weight update. Ref[14] have learned a network that can detect a 1-bit message from a image sub-block. Experimental results show that this learned network is an extension of the frequency domain that is widely used in existing watermarking scheme. But this method did not mention the keys and its capacity is not high.

III. WATERMARKING PROPOSED METHOD

A. Watermark sequence generating and Embedding algorithm

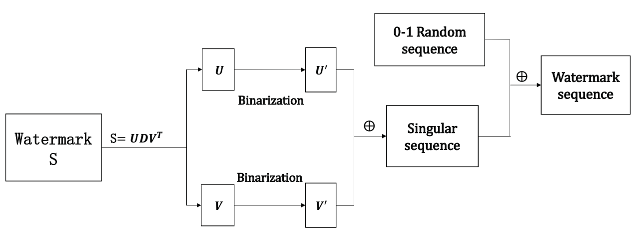


Fig. 1. The process of generating watermark sequence

As shown in figure 1, Firstly, using singular value decomposition, the watermark is decomposed into singular values and singular vectors; Secondly, the singular vectors are barbarized and merged to obtain singular sequences; Finally, the singular sequences are scrambled to obtain the watermark sequence. If the size of the watermark is 512×512, the length of the watermark sequence is 512^2, that is, 262144. The specific process of watermark sequence generating is as follows:

Firstly, according to the $S = U * D * V$, the S is decomposed, and left singular matrix U, the diagonal matrix D and the right singular matrix V are obtained. Secondly, binarization of U and V, respectively, as shown in formula:

$$
\begin{aligned}
U'(i) &= \left\{ \begin{array}{l} 1, U(i) > u \\ 0, U(i) \leq u \end{array} \right\} (u = \text{median}(U), i = 1, 2, .., N) \\
V'(i) &= \left\{ \begin{array}{l} 1, V(i) > v \\ 0, V(i) \leq v \end{array} \right\} (v = \text{median}(V), i = 1, 2, .., N)
\end{aligned} \tag{1}
$$

Where N is the length of U, $[\cdot] = median(\cdot)$ is the function of getting the mean value in Matlab, u is the mean value of the vector U while v is the mean value of the vector V. The following operations are performed for U' and V', according to formula $G(i) = U(i) \oplus V(i) \quad (i = 1, 2, ..., N)$, where $\oplus$ is the operator of XOR. Finally, based on the key of k, generates pseudo-random numbers K whose length is N, and performs the operation like formula $G(i) = U(i) \oplus V(i) \quad (i = 1, 2, ..., N)$ on K and G, watermark sequence W is obtained.

For embedding algorithm, considering the human eye's visual characteristics, we choose to embed watermark information in the wavelet transform domain of the host image. A DWT transform of the host image is carried out to obtain four strip LL, LH, HL, HH. The singular values of the watermark are then embedded into the HH strip. The LL strip is decomposed by multilevel wavelet transform, and the watermark sequence is embedded into the low frequency component and high frequency component of LL strip. This section takes 512×512 watermark and 512×512 host image as an example, and carries out the 4 level wavelet decomposition to the LL strip. The specific process called the DWT-SVD-embedding of embedding algorithm is as follows:

We perform the operations following as $[LL, LH, HL, HH] = dwt2(A)$ on host image A and LL, LH, HL, HH and A is respectively low frequency component, level of detail components, vertical and diagonal detail component detail component. Then, calculate singular value decomposition of watermark S , so that the left singular vector U_w, diagonal matrix S_w, right singular vector V_w are obtained; Perform the following operations on the result diagonal matrix S_h and S_w of singular value decomposition of HH :$diag(S_h) \leftarrow diag(S_w)$ and $[\cdot] = diag(\cdot)$is the function of the diagonal element of the fetch matrix in Matlab, that is, the diagonal element of $S_h$is replaced by the diagonal element of S_w; Perform inverse singular value decomposition on S_h:$HH' = U_h S_h V_h$; The LL is decomposed into four levels by formula $[LL, LH, HL, HH] = dwt2(A)$, and $LL_4 \ and \ HH_4$ are obtained. The following operations are performed on $LL_4 \ and \ HH_4$:

$$
\left\{ \begin{array}{l} LL_4 = reshape(LL_4, 1, length(LL_4)^2) \\ HH_4 = reshape(HH_4, 1, length(HH_4)^2) \\ LLHH = [LL_4, HH_4] \end{array} \right. \tag{2}
$$

Where $[\cdot] = length(\cdot)$is the function of the maximum length of all dimensions of the vector in the Matlab, and $[\cdot] = reshape(\cdot, \cdot, \cdot)$ is the indefinite dimension function.

Then, we can separate the decimal part from the integer part of each bit in the $LLHH$ for the following operations:

$$
\left\{ \begin{array}{l} LLHH = abs(LLHH) \\ integer = fix(LLHH) \\ fraction = LLHH - integer \end{array} \right. \tag{3}
$$

Where, $[\cdot] = abs(\cdot)$is the absolute function in Matlab, $[\cdot] = fix(\cdot)$is the rounding (closer to 0) function in Matlab, $integer$ is the integer part of $LLHH$ $fraction$, $LLHH$ is the decimal part.

Finally, we replace W which is the result of the watermark sequence generating with tenth bits of $integer$ (from high to low bit), get $integer'$, and do the following for $integer'$:$\mathrm{LLHH_{modi}} = \mathrm{integer}' + \mathrm{fraction}$. According to the formula (2), the inverse operation is performed, and the $LL4$ and $HH4$ after embedding the watermark sequence are obtained from the $LLHH_{modi}$, corresponding

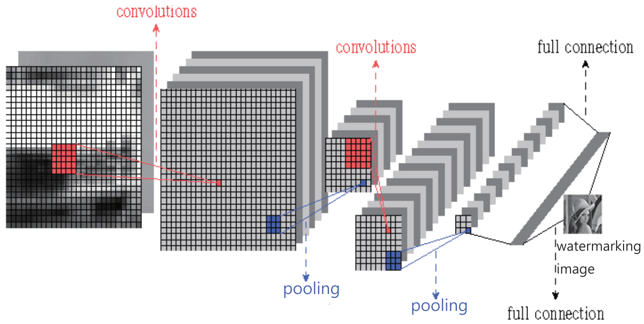


Fig. 2. The structure of the convolution neural network

to $LL4_{modi}$ *and* $HH4_{modi}$ respectively. Apply 4 level inverse wavelet decomposition to $LL4_{modi}$ and $HH4_{modi}$, the embedded watermark sequence after the LL, that is, LL'. The LL', HH', $HL$and LH are decomposed by inverse wavelet, and the watermark image WA is obtained:$WA = idwt2(LL', LH, HL, HH')$ where $[\cdot] = idwt2(\cdot,\cdot,\cdot,\cdot)$ is a function of two-dimensional inverse discrete wavelet transform in Matlab.

B. Image Watermark Based on CNN

As show in algorithm 1, we divide watermarking into blocks. In this section, the watermarking is divided into blocks sized 1616. Based on the relation between the divided watermarking and host image, we construct the convolution neural network as shown in Fig. 2. Using different key k, we generate different watermarking image as training data for network to adjust the weights. Do the same division operating to host image as watermarking, make it as convolution neural networks input, then adjust the watermarking images pixel value according to networks output. Above all, we finish the whole construction of convolution neural network.

IV. EXPERIMENT RESULTS

We use 5 original 512×512 gray scale images, named: "House", "Man", "Lena", "Pepper", "Airplane". They are shown in Fig 3. A logo watermark is a 512×512 binary image shown in Fig 4.

Fig. 3. From left to right is the original images of "House", "Man", "Lena", "Pepper" and "Airplane"

Fig. 4. The logo image

Algorithm 1: Image watermark based on convolution neural network

Input: watermarking S, host image A, training times $numOfKeys$

Output: watermarking image WA'

1: Set the basic parameters of the convolution neural network, get network cnn, $opts$;

2: Divide S by random walks for image segmentation[15], get blocks $train\_x$, which size is $16 \times 16 \times 512$;

3: Use $train\_x$ as the training input, A as the training output, initialize cnn:

$$\mathrm{cnn} = \mathrm{cnnsetup(cnn, tran\_x, A)}; \tag{4}$$

In (4), $[\cdot] = cnnsetup(\cdot,\cdot,\cdot)$ is the function to initialize network weights in convolution neural network package.;

4: Divide A, get blocks $test\_x$, which size is $16 \times 16 \times 512$;

5: **for** $key = 1 \rightarrow m + numOfKeys$ **do**

  $WA \leftarrow$ DWT-SVD-EMBEDDING(A, S) ;

  $\mathrm{cnn} \leftarrow \mathrm{cnntrain(cnn, train\_x, WA, opts)}$;

  $\mathrm{labels} \leftarrow \mathrm{cnnpredict(cnn, test\_x)}$;

  $WA \leftarrow WA. * labels$;

6: $WA' \leftarrow WA$

7: **return** WA'

In order to evaluate how much robust the proposed method is, the watermarked image is subjected to different types of attacks[10], [11], [14]. The attacks applied are Average Filtering, Motion-Blur, Sharpen and so on. Then, the Peak Signal to Noise Ratio (PSNR), Normalized Correlation (NC) and the correlation coefficient (CC) of the extracted watermark with respect to the original watermark are calculated.

Peak Signal to Noise Ratio (PSNR) is computed to analyze the concealing effect of the watermark. It is calculated as the ratio between the maximum power of the original image and the power of unwanted noise which is added to the image (which will affect the exactness of its representation). The formula to find out PSNR is shown in (5).

$$PSNR = 10\log_{10}\frac{M^2}{\frac{1}{m\times n}\sum\limits_{i=0}^{m-1}\sum\limits_{j=0}^{n-1}[I(i,j) - K(i,j)]^2} \tag{5}$$

Where, M is the power of the signal, $I(i,j)$ is the original watermark and $K(i,j)$ is the extracted watermark. Bigger the PSNR value better the watermark conceals.

The robustness of the proposed algorithm is analyzed by using Normalized Cross Correlation (NC). It is a metric to evaluate the degree of similarity (or dissimilarity) between two compared images. The original watermark and the extracted watermark are compared. The equation to compute NC is

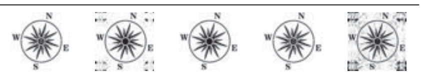
TABLE I

COMPARISON THE ROBUSTNESS (CC)

Attack	House Ref[12]	House Ref[10]	House Ours	Lena Ref[12]	Lena Ref[10]	Lena Ours	Pepper Ref[12]	Pepper Ref[10]	Pepper Ours	Man Ref[12]	Man Ref[10]	Man Ours	Airplane Ref[12]	Airplane Ref[10]	Airplane Ours
Rotation (15°)	0.7540	0.7570	0.9682	0.6474	0.7350	0.9737	0.7112	0.7862	0.9638	0.6624	0.7887	0.9667	0.6477	0.7615	0.9664
Resize ($512 \to 1024 \to 512$)	0.6670	0.7614	0.979	0.6918	0.7523	0.9837	0.6576	0.7860	0.9742	0.6971	0.7542	0.9768	0.7066	0.7387	0.9879
JPEG Compression (QF =20)	0.7300	0.7788	0.9847	0.6765	0.7554	0.9857	0.6395	0.7465	0.9820	0.6541	0.7562	0.9729	0.6464	0.7378	0.9867
Average Filtering (3×3)	0.7190	0.7399	0.9797	0.7144	0.7885	0.9907	0.6854	0.7894	0.9846	0.7316	0.7665	0.9719	0.6757	0.7356	0.9853
Median Filtering (5×5)	0.7250	0.7461	0.979	0.6418	0.7532	0.9865	0.7282	0.7813	0.9865	0.6315	0.7329	0.9759	0.7135	0.7346	0.9879
MotionBlur	0.7450	0.7421	0.9612	0.7187	0.7709	0.9530	0.6395	0.7759	0.9480	0.6714	0.7709	0.9625	0.7118	0.7591	0.9524
Histogram Equalization	0.6490	0.7648	0.9256	0.6746	0.7498	0.9250	0.7230	0.7767	0.9260	0.6494	0.7433	0.9491	0.6457	0.7732	0.9252
Sharpen	0.7320	0.7430	0.9913	0.6447	0.7541	0.9613	0.7188	0.7695	0.9607	0.6896	0.7788	0.9602	0.6456	0.7369	0.9607
Contrast Adjustment	0.7280	0.7878	0.9586	0.7172	0.7785	0.9250	0.6938	0.7612	0.9260	0.6664	0.7582	0.9491	0.6716	0.7588	0.9252
Cropping ($1/4$) Area Remaining	0.6920	0.7504	0.9804	0.6387	0.7625	0.9921	0.6552	0.7626	0.9948	0.6435	0.7811	0.9806	0.6552	0.7797	0.9888

TABLE II

COMPARISON THE ROBUSTNESS (NC)

Attack	House Ours	Lena Ours	Pepper Ours	Man Ours	Airport Ours
Rotation(45) 	0.9256	0.9047	0.9873	0.9406	0.9037
JPEG Compression(QF =10) 	0.9566	0.9681	0.9346	0.9544	0.957
JPEG Compression(QF =40) 	0.9584	0.9667	0.9515	0.954	0.9556
JPEG Compression(QF =60) 	0.9791	0.9847	0.9575	0.9019	0.9228
JPEG Compression(QF =90) 	0.9508	0.9457	0.9537	0.9194	0.9472
JPEG Compression(QF =100) 	0.9508	0.9457	0.9537	0.9194	0.9472

given in (6).

$$NC = \frac{\sum_{i=1}^{M_1}\sum_{j=1}^{M_2} W(i,j)W'(i,j)}{\sqrt{\sum_{i=1}^{M_1}\sum_{j=1}^{M_2}[W(i,j)]^2}\sqrt{\sum_{i=1}^{M_1}\sum_{j=1}^{M_2}[W'(i,j)]^2}} \quad (6)$$

Where, $W(i,j)$ is the original watermark and $W'(i,j)$ is the extracted watermark. The value of NC is between 0 and 1. As the value increases, the method will be more robust.

Moreover, the correlation coefficient is used to compute the difference between the original watermark and the extracted watermark. The correlation coefficient is defined :

$$CC = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}[W(i,j)-\hat{W}][W'(i,j)-\hat{W}']}{\sqrt{[\sum_{i=1}^{M}\sum_{j=1}^{N}[W(i,j)-\hat{W}]^2][\sum_{i=1}^{M}\sum_{j=1}^{N}[W'(i,j)-\hat{W}']^2]}} \quad (7)$$

Where W and W' are the original image and the watermarked image, M and N are the height and the width of the original image and the watermarked image. $\hat{W}$ and $\hat{W}'$ are the mean values of the original watermark and the extracted watermark.

Embedded watermarks which is a logo that are better than Ref[12] method and Ref[10] method about the capacity in Table III method. Additionally, our method is also higher than that two methods about the imperceptibility (higher PSNR). The result of the Table I shows that our method is more robust than Ref[12] method and Ref[10] method. However, our method is more efficient than others.

TABLE III

COMPARISON CAPACITY AND PSNR

Image name	Method	Capacity	PSNR
House	Ref. [12]	512*512	17.38
	Ref. [10]	512*512	20.6
	Ours	512*512	38.5659
Lena	Ref. [12]	512*512	18.52
	Ref. [10]	512*512	21.56
	Ours	512*512	39.9064
Pepper	Ref. [12]	512*512	23.32
	Ref. [10]	512*512	25.2
	Ours	512*512	38.5365
Man	Ref. [12]	512*512	20.43
	Ref. [10]	512*512	2.73
	Ours	512*512	35.3153
Airplane	Ref. [12]	512*512	18.03
	Ref. [10]	512*512	24.19
	Ours	512*512	39.415

Additionally, in the Table II, watermarked images are attacked by Matlab 7.1. As a result the NC obtain high value from 0.9019 to 0.9873 . This is our valuable contribution in comparison with Ref[12] method and Ref[10] method.

V. CONCLUSION

In this paper, we proposed a robust blind watermarking technique with high capacity, robustness and security using CNN and DWT-SVD. Experiments have yielded higher PSNR and NC values than comparative methods for test images. And our algorithm ensures the security of the watermarks in the embedding and the extraction process. The strength of the proposed scheme is demonstrated through successful watermark detection after various attacks. Moreover, by comparison, our algorithm is dominated at all.

ACKNOWLEDGMENT

This paper was supported in part by Science & Technology Pillar Program of Hubei Province under Grant (#2014BAA146), Nature Science Foundation of Hubei Province under Grant (#2015CFA059), Science and Technology Open Cooperation Program of Henan Province under Grant (#152106000048).

REFERENCES

[1] N. Liu, H. Li, H. Dai, D. Guo, and D. Chen, "Robust blind image watermarking based on chaotic mixtures," *Nonlinear Dynamics*, vol. 80, no. 3, pp. 1329–1355, 2015.

[2] S. Bekkouch and K. M. Faraoun, "Robust and reversible image watermarking scheme using combined dct-dwt-svd transforms," *Journal of Information Processing Systems*, vol. 11, no. 3, pp. 406–420, 2015.

[3] A. Mishra, C. Agarwal, A. Sharma, and P. Bedi, "Optimized gray-scale image watermarking using dwt-svd and firefly algorithm," *Expert Systems with Applications*, vol. 41, no. 17, pp. 7858–7867, 2014.

[4] J. Ouyang, G. Coatrieux, B. Chen, and H. Shu, "Color image watermarking based on quaternion fourier transform and improved uniform log-polar mapping," *Computers & Electrical Engineering*, vol. 46, pp. 419–432, 2015.

[5] W. H. Lin, Y. R. Wang, S. J. Horng, T. W. Kao, and Y. Pan, "A blind watermarking method using maximum wavelet coefficient quantization," *Expert Systems with Applications*, vol. 36, no. 9, pp. 11509–11516, 2009.

[6] M. Vafaei, H. Mahdavi-Nasab, and H. Pourghassem, "A new robust blind watermarking method based on neural networks in wavelet transform domain," *World Applied Sciences Journal*, vol. 22, no. 11, pp. 1572–1580, 2013.

[7] N. M. Makbol and B. E. Khoo, "A new robust and secure digital image watermarking scheme based on the integer wavelet transform and singular value decomposition," *Digital Signal Processing*, vol. 33, p. 134147, 2014.

[8] G. E. Hinton, "To recognize shapes, first learn to generate images," *Progress in Brain Research*, vol. 165, no. 6, pp. 535–547, 2007.

[9] M. Kim, D. Li, and S. Hong, "A robust digital watermarking technique for image contents based on dwt-dfrnt multiple transform method," *International Journal of Multimedia & Ubiquitous Engineering*, vol. 9, no. 1, pp. 369–378, 2014.

[10] T. H. Nguyen, D. M. Duong, and D. A. Duong, "Robust and high capacity watermarking for image based on dwt-svd," in *IEEE Rivf International Conference on Computing & Communication Technologies - Research, Innovation, and Vision for the Future*, pp. 83–88, 2015.

[11] A. K. Gupta and M. S. Raval, "A robust and secure watermarking scheme based on singular values replacement," *Sdhan*, vol. 37, no. 4, pp. 425–440, 2012.

[12] M. Makhloghi, F. Akhlaghian, and H. Danyali, "Robust blind dwt based digital image watermarking using singular value decomposition," *International Journal of Innovative Computing Information & Control Ijicic*, vol. 8, no. 7, pp. 219–224, 2011.

[13] C. Farabet, C. Couprie, L. Najman, and Y. Lecun, "Learning hierarchical features for scene labeling.," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 35, no. 8, pp. 1915–1929, 2013.

[14] S. M. Mun, S. H. Nam, H. U. Jang, D. Kim, and H. K. Lee, "A robust blind watermarking using convolutional neural network," 2017.

[15] L. Grady, "Random walks for image segmentation.," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 28, no. 11, p. 1768, 2006.