Digital image watermarking using deep learning

Article *in* Multimedia Tools and Applications · May 2023

DOI: 10.1007/s11042-023-15750-x

2 authors:

Himanshu Kumar Singh
National Institute of Technology Patna

Amit Singh
Punjab Technical University

Multimedia Tools and Applications
https://doi.org/10.1007/s11042-023-15750-x

Digital image watermarking using deep learning

Himanshu Kumar Singh[1] · **Amit Kumar Singh**[1]

Received: 21 February 2023 / Revised: 31 March 2023 / Accepted: 24 April 2023

Abstract
At present, watermarking techniques play an important role in protecting digital images. To date, many classical watermarking schemes have been developed to protect images based on spatial and transform domains. However, classical watermarking schemes are less resilient to many attacks. Recently, deep learning-based watermarking made a significant contribution to image content security and received attention for various popular applications. In this paper, we use convolutional neural networks (CNNs) to propose an interesting watermarking technique for digital images. Initially, latent features of cover and secret images are extracted using an encoder network and later concatenated to generate a marked image. On the receiver side, a denoising autoencoder network is used to remove noise variations from the received image and later to extract the secret mark image using a CNN. Our technique not only imperceptibly hides an image inside a cover image but also outperforms other state-of-the-art schemes in terms of visual quality and robustness according to simulation results and performance comparisons.

Keywords Watermarking · Deep learning · Digital image · Autoencoders

1 Introduction

In this big data era, digital images are becoming increasingly important in various fields for their potential applications in medicine, social media, forensics, cinematography, education and other fields. These images may contain private and sensitive information about the content owner. Unauthorised access to these sensitive images could lead to more serious issues, such as privacy leakage, copyright flouting and interference with doctors' diagnoses [2]. Digital image security is critical for this reason. At present, watermarking techniques play an important role in protecting digital images. Image watermarking hides copyright marks inside cover images, making them imperceptible and robust at the same time. In classical watermarking (Fig. 1), the embedding of copyright marks is done either

✉ Amit Kumar Singh
amit.singh@nitp.ac.in

Himanshu Kumar Singh
himanshus.ph21.cs@nitp.ac.in

[1] Department of Computer Science & Engineering, National Institute of Technology Patna, Patna, Bihar 800005, India

Published online: 13 May 2023

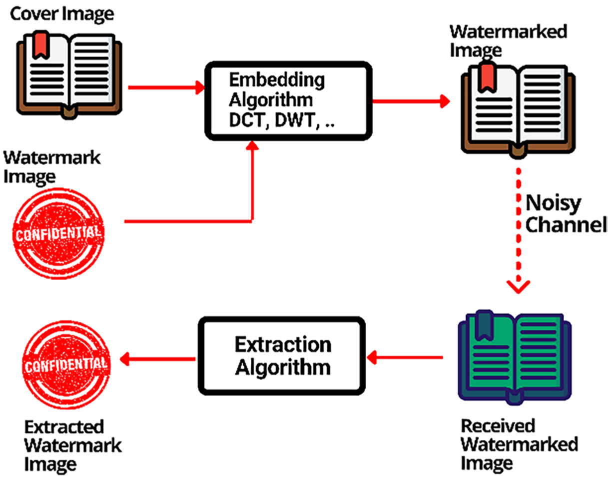


Fig. 1 Overview of classical watermarking

by directly modifying the pixel value or by modifying the transform coefficient of the cover image. Compared with the spatial domain scheme, the transform domain scheme provides better robustness and flexibility [13]. However, classical watermarking schemes are less resilient to attacks and their applications are narrow [1]. Therefore, the effective robust watermarking method for digital images deserves an in-depth investigation.

Recently, deep learning-based watermarking made a significant contribution to image content security and received more attention for various popular applications [21]. The following are the main benefits of using deep learning for watermarking: (a) locating the ideal embedding position within the cover media; (b) determining the ideal embedding strength that offers a balanced trade-off between imperceptibility and robustness; (c) providing attack simulation for effective watermark extraction; and (d) minimising errors and noise for obtained watermarks [3]. There are three main criteria to evaluate any image watermarking method: imperceptibility, robustness and watermark capacity [18]. Generally, robustness is the most important performance marker of the watermarking system. While performing watermarking, the original media should not be visibly distorted after concealing the hidden data.

Motivated by the recent success of deep learning, we have used convolutional neural networks (CNN) to propose an interesting watermarking technique for digital images. An autoencoder-based embedder network is developed, which maintains the high visual quality of the marked images. Additionally, a denoising network is used to propose an extractor network to remove noise variations from the possibly distorted marked image before extraction, which improves the robustness of the watermarking scheme. Initially, the learning ability of the deep learning network is utilised to automatically learn and generalise the watermarking algorithm, providing an automated system without the need for domain knowledge. After this, the embedder and extractor network are trained in an unsupervised manner to reduce human intervention. Compared with conventional methods, the proposed method is more robust and imperceptible while embedding different sizes of mark data.

The rest of this paper is organized as follows. Section 2, introduces the related work and compares them with our work. In Section 3, the proposed watermarking technique in terms of embedding and extraction networks is described in detail. Experimental details are reported in Sectiom 4. Finally, the paper is concluded in Section 5.

2 Related work

Deep learning has recently shown great success in the image processing field [5, 15]; therefore, it could be an excellent option for watermarking applications. In 2020, Bagheri et al. [4] used deep learning to identify the appropriate location for embedding the mark. A deep network mask region-based CNN was developed and trained on the Common Objects in Context dataset. Although the experimental results demonstrated good transparency and robustness of the marked data, the security of the watermark needs to be further investigated. Wei et al. [23] described a robust watermarking scheme by using a cycle variational autoencoder. The network learned to embed and extract 1-bit mark images, improving their visual quality. However, its watermark capacity was low, limiting the use of the method for practical applications. Ge et al. [9] designed a document image watermarking scheme by using an encoder-decoder network. The scheme used the noise layer and watermark expansion approach to improve resilience against attacks. However, the scheme was embedded-strength dependent and did not perform well against JPEG attacks. Zhong et al. [25] proposed a hiding scheme based on the convolutional network. Two different networks (i.e., embedder and extraction networks) were used to embed and extract the watermark. Additionally, to improve robustness, a fully connected invariant network was used to learn the noise variations in the watermarked image. However, the end-to-end training of the network led to information loss.

Ding et al. [7] designed a watermarking scheme using a deep neural network. Initially, up-sampling was applied on the cover and mark images using the transpose convolutional network. After that, a blender network was used to blend the watermark and cover images. Subsequently, a sampler was used to obtain a marked image. The extractor network was composed of a convolutional block to extract the watermark. Though this scheme achieved high invisibility, it did not always produce good resilience against attacks such as JPEG, median and low-pass filtering and rotation attacks. A blind DCT-SVD-based watermarking is described by Wang et al. [22]. Initially using the median filter, the cover image was enhanced to improve the robustness of the watermark. Later, without altering the cover picture, Region-based CNN was used to map the association between the watermark and cover images. The non-embedding technique improved the robustness of the watermark but also increased the complexity of the technique. Zheng et al. [24] designed a method to investigate the imperceptibility and robustness of the watermark. Initially, the cover media was transformed into different bands using discrete wavelet transform, and then the watermark was inserted into the high bands of the cover media. Then, the transformation was applied to the low bands by wavelet transformation, where the watermark sequence was embedded into the selected low bands. Later, a CNN network was used to extract the watermark from the cover image. Islam et al. [10] proposed a reliable watermarking method utilising an artificial neural network (ANN). The watermark was embedded using the lifting wavelet transform (LWT) and randomised coefficient. The selected sub-band coefficient was first randomised using a key after the cover picture had been modified using the LWT. Later, using a different key, the randomised coefficient was used to obtain the randomised blocks. The chosen sub-randomised band's block was then used to incorporate the watermark. ANN was utilised for watermark detection and later extracted using the inverse of the embedding procedure.

In [16], Mahapatra et al. proposed a convolutional autoencoder-based image watermarking scheme. The watermark was embedded by concatenating the watermark and cover images using the encoder-decoder network. A deep neural network was used to capture the invariant feature from the marked image and later reconstructed using a transposed convolutional block to obtain the watermark. The experimental results showed the robustness of the scheme, but the network was trained on the noiseless marked image, which was not able to differentiate between noise variation and watermark variation, leading to extracting the noisy information.

The analytical comparison of our proposed technique with the recent state-of-the-art technique is shown in Table 1. Although the above deep learning-based watermarking approaches were developed to provide copyright protection and authentication of media, most of them have limited robustness and visual quality. To address the above issues, we have utilised the convolutional autoencoder framework in the embedding network to improve the visual quality. Subsequently, a denoising network was used in the extractor network to preserve the watermark information in the marked image. The upcoming section presents the proposed watermarking scheme in detail.

3 Description of the proposed watermarking

The proposed watermarking technique is composed of two stages (as illustrated in Fig. 2): (1) Embed the secret data into the cover image by an embedding network and (2) extract the secret data from the marked image. The following sections provide further detail about the proposed scheme.

3.1 Embedding network

Given the cover (C) and mark (W) images, the latent features of both C and W are computed, which are then concatenated via embedder network μc and μw, respectively, as shown in Fig. 3. Inversely, the decoder network (σ_W) learns a decoding function to decode the concatenated feature to obtain the marked image (M). Here, the latent representation of cover and mark images are denoted as C_Z and W_Z, respectively. The encoder progressively decreases the size of the cover image feature map to make it equal to the mark feature map so that the feature maps of W_Z and C_Z can be concatenated. Later, the decoder progressively increases the feature map to obtain the marked image. The specific steps for embedding the secret image are described in Algorithm 1.

3.2 Extractor network

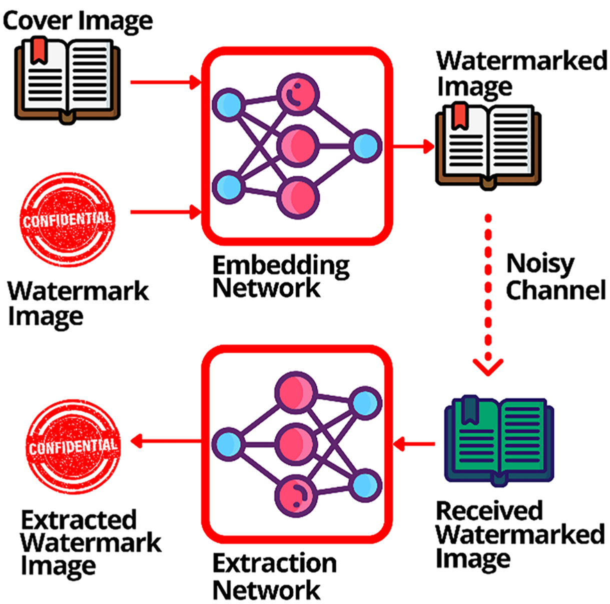
The extraction network is composed of a denoising encoder-decoder network along with a convolutional block, as shown in Fig. 4.

The extractor network extracts the embedded watermark image from the watermarked image. Initially, a denoising autoencoder network is used to reduce noise effect (if any) from the received data at receiver side. Later, the encoders are used to obtain the latent feature from the denoised image and the cover image. Here, the extracted latent feature of the cover image is subtracted from the marked latent feature to obtain the residual of the marked image. Subsequently, the obtained residual features are flattened to 16,384 network parameters. The CNN block is used to make the flattened features dense and later concatenated and reshaped. Finally, the decoder network is used to obtain watermark images by progressively increasing the reshaped feature map. The specific steps for extracting the secret image are described in Algorithm 2.

Table 1 An analytic comparison between recent work and the proposed scheme

Method	DL model used	Model role	Noticed limitations
Bagheri et al. [4]	CNN	Calculation of embedding strength	-Security of the mark data needs to be analysed.
Wei et al. [23]	Cycle variational autoencoder	Embedding and extraction of watermark	-Limited capacity of the scheme
Ge et al. [9]	Autoencoders	Document watermarking	-Dependant on embedding strength -Limited applicability.
Zhong et al. [25]	CNN	Embedding and extraction of watermark	-Limited capacity of the scheme -Information loss due to end-end training.
Ding et al. [7]	CNN	Embedding and extraction of watermark	-Limited robustness analysis. -Poor performance against most of the considered attacks
Wang et al. [22]	CNN	Watermark embedding	-High complexity in terms of embedding and extraction cost.
Zheng et al. [24]	CNN	Watermark embedding	-Scheme complexity needs to be analysed.
Islam et al. [10]	ANN	Watermark detection	-Low embedding capacity -Limited robustness analysis
Mahapatra et al. [16]	Autoencoder	Watermark embedding and extraction	-Extraction of noisy information.
Ours	Denoising autoencoder, DNN	Watermark embedding and extraction	-May not be appropriate for dual watermarking

Fig. 2 Overview of the proposed model



4 Experiments and analysis

This section presents a series of simulation results to prove the effectiveness of our proposed scheme. To evaluate the embedding and recovering performance of our scheme, we used three metrics to measure the quality of the marked image and the recovered mark image, including the peak signal-to-noise ratio (PSNR) [1, 19], structural similarity index measure (SSIM) [1, 19] and normalised correlation (NC) [1]. The following sections provide further details on the results and analysis of the proposed scheme.

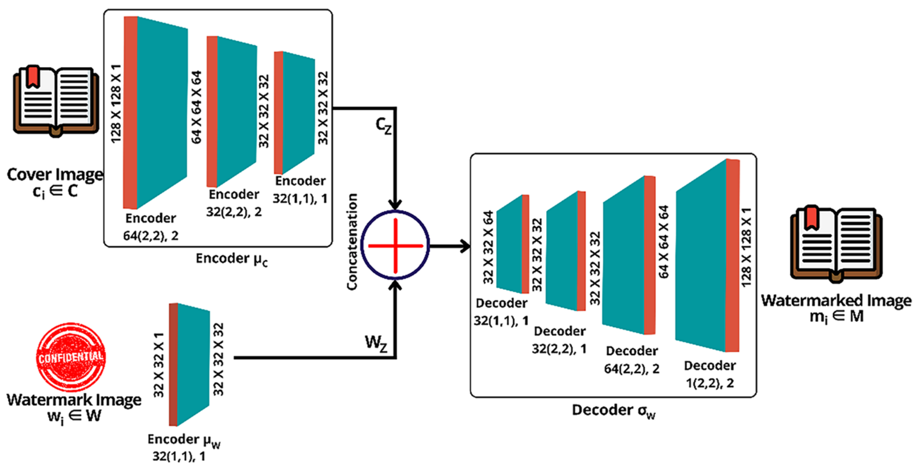


Fig. 3 Detailed architecture along with network configuration of the embedder network

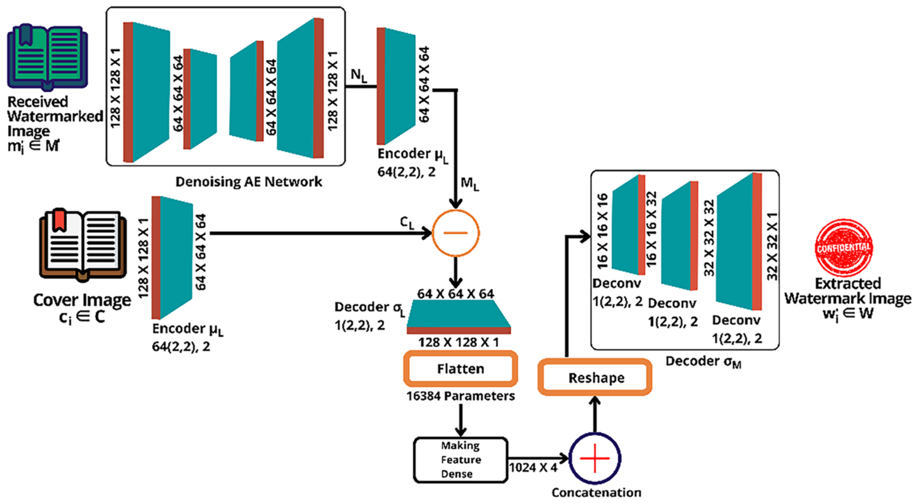


Fig. 4 Detail architecture along with network configuration of extraction network

4.1 Preparation of datasets

For the training and testing of the watermarking network, Cats and dogs [11] and CIFAR [12] datasets were used as the cover and mark images, respectively. Some samples from each dataset are shown in Fig. 5a and b. The cover image dataset contained 10,000 training samples and 8,000 testing samples. The watermark dataset contained 1,000 training samples and 600 testing samples. A noisy watermarked dataset (M') was prepared using the data augmentation process [17] for the training of the extractor network. The testing samples were not used in the training process to demonstrate the generalising and learning capabilities of the proposed scheme.

4.2 Training and testing details

The proposed watermarking technique was trained in two phases (i.e., embedding network training and extraction network training). The mean squared error was used to compute the loss function during the training of both networks. The hyperparameter details of both networks are shown in Table 2.

For the training of the embedding network, the Adaptive Moment Estimation optimiser [8] was used because of its ability to continuously learn after each epoch. The training and validation of the embedding network are shown in Fig. 6, where the loss (L_1) (Eq. 1) during each epoch is presented. The smaller gap between the training and validation losses indicates that the model cannot be categorised as overfitting. All the layers of the network applied the rectified linear unit (ReLU) as the activation function except for the output layer, which used the sigmoidal function to limit the range to (0,1). During the testing phase, the PSNR and the SSIM were used to evaluate the fidelity of the marked image.

$$L_1 = MSE(C, M) \tag{1}$$

Algorithm 1 Embedding algorithm

Input: Cover images C, Watermark images W

Output: Watermarked image M

1. **1. Initialize:**
   B: Number of batches ← 32
   η: Learning rate ← 0.001
   e: *Number of epochs* ← 300
   α: Number of kernels
   β: Kernel size
   2. Reading data:
   Load Dataset C
   Load Dataset W
   3. Pre-processing image dataset:
   Resize (Grayscale(C), $\mathbf{128} \times \mathbf{128} \times \mathbf{1}$)
   Resize (Grayscale(W), $\mathbf{32} \times \mathbf{32} \times \mathbf{1}$)
   4. Make encoder for cover image & extract features:
   C_Z ← *Encoder* μ_C *(C, α, β)*
   5. Make encoder for watermark image & extract features:

   W_Z ← *Encoder* μ_W(W, *α*, *β)*

   6. Concatenate features:

   M_Z ← *Concatenate* (C_Z, W_Z)

2. **7. Make model Decoder on concatenated features:**
   Decoder σ_W (M_Z, *α*, *β*)

3. **8. Compile model:**
   A: Load optimizer ← *Adam (η)*
   MSE: *Mean Squared Error*

   Embedder ← *compile* (μ_C, μ_W, σ_W, *A*, *MSE*)

4. **9. Train and Test Model:**
   Training:
   for 0 to *e* do
   for 0 to *B* do
   Step 1: Input images in the model:
   M_i ← *Embedder* (C, W)
   Step 2: Calculate loss:
   L_1 ← *MSE* (*C*, M_i)
   Step 3: Apply Adam optimizer:
   Calculate gradients:
   G_1 ← *A* (L_1, *α*, *β*)
   Step 4: Apply gradients on the model (update weights):
   α, *β* ← *A* (G_1, *α*, *β*)
   end for
   end for
   Testing:
   M ← *Embedder* (*TestData_C, TestData_W)*
   10. Calculate:
   PSNR (*C*, M)
   SSIM (*C*, M)

Algorithm 2 Embedding algorithm

Input: Watermarked images M', Cover images C

Output: Extracted watermarks W'

1. **1. Initialize:**
B: Number of batches ← **32**
η: Learning rate ← **0.001**
e: *Number of epochs* ← **300**
α: Number of kernels
β: Kernel size
2. Reading data:
Load Dataset **M'**
Load Dataset **C**
3. Pre-processing image dataset:
Resize **(Grayscale(C), $128 \times 128 \times 1$)**
Resize **(Grayscale(M'), $128 \times 128 \times 1$)**
4. Make denoising AE for noisy marked images:
N_L ← *Denoising AE (*M', *α*, *β*)
5. Make encoder for cover image & extract feature:
C_L ← *Encoder μ_L (C, α, β)*
6. Make encoder for watermarked images & extract features:

M_L ← *Encoder μ_L* (N_L, *α*, *β)*

7. Subtract features:
S_Z ← *Subtract* (M_L, C_L)

2. **8. Make Decoder for Subtracted features:**
D_L ← *Decoder* σ_L (S_Z, *α*, *β*)

9. Flatten the decoded feature: 16384 parameters
10. Making feature dense:
features = *empty list ()*
for j from 1 to 4:

$$y_j = DNN\ (D_L)$$

$$y_j = batchnorm(y_j)$$
$$y_j = relu(y_j)$$

$$Y = features.append(y_j)$$

end for

11. Y = *reshape***(Y) to 16x16x16**
12. Apply decoder: σ_M (Y, *α*, *β*)

13. Compile model:

Extractor ← *compile* (*μ_L*, σ_L, σ_M, *A*, *MSE*)

14. Train and Test Model:

Training:

for 0 to *e* do
for 0 to *B* do

Step 1: Input images in the model:

P_i ← *Extractor* (*C, M'*)

Step 2: Calculate loss:

L_2 ← *MSE* (*W*, *P_i*)

Step 3: Apply Adam optimizer:
Calculate gradients:

G_2 ← *A* (L_2, *α*, *β*)

Step 4: Apply gradients on the model (update weights):

α, *β* ← *A* (G_2, *α*, *β*)

end for
end for

Testing:
W' ← *Extractor* (*TestData_C, TestData_M')*

15. Calculate:

NC (W, W')

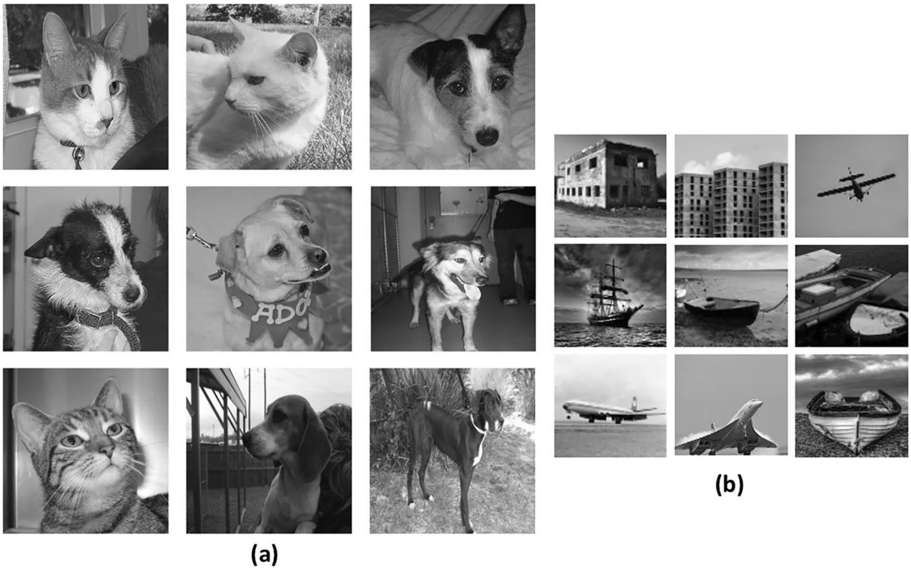


Fig. 5 **a** Samples from the cover image dataset and **b** samples from the watermark image dataset

The testing PSNR was 44.48 dB and the SSIM was 0.9997, indicating the high fidelity of the marked images. Therefore, embedded information was unnoticeable to the human eye. A few of the testing examples are shown in Fig. 7.

The extraction network was trained using the noisy watermarked images, which allowed the network to learn the noise variation. This was done so that the extraction network could extract the watermark image even in cases where the watermarked image contained some level of noise. The training and validation of the extraction network are shown in Fig. 8, where the values of the loss (L_2) (Eq. 2) during each epoch is presented. The smaller gap between the training and validation loss indicates the model learning performance for the watermark extraction. The ADAM optimiser was used for each epoch and all layers except for the output layer, which used the ReLU activation function and the sigmoidal function. During the testing phase, the NC value was determined to evaluate the quality of the extracted watermark image. On the test dataset, the obtained NC score was 0.9996. A few of the test images are illustrated in Fig. 9.

$$L_2 = MSE(W, W') \tag{2}$$

Table 2 Hyperparameters used in watermarking network

Hyperparameters	Embedding network	Extraction network
Optimizer	ADAM	ADAM
Learning rate	0.0001	0.0001
Beta 1	0.9	0.9
Beta 2	0.999	0.999
Loss	Mean Squared Error	Mean Squared Error
Epochs	135	185

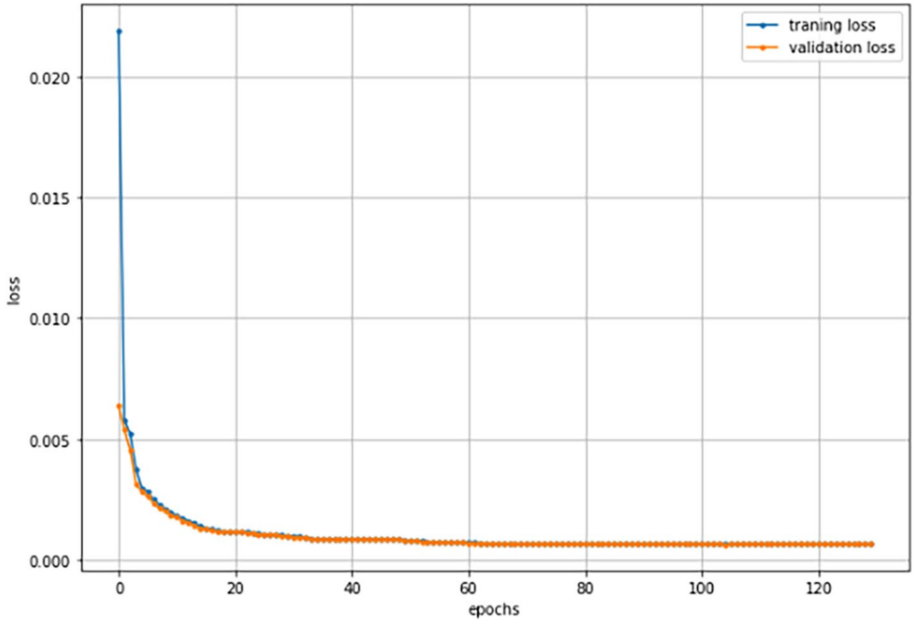


Fig. 6 Training and validation loss for embedding network

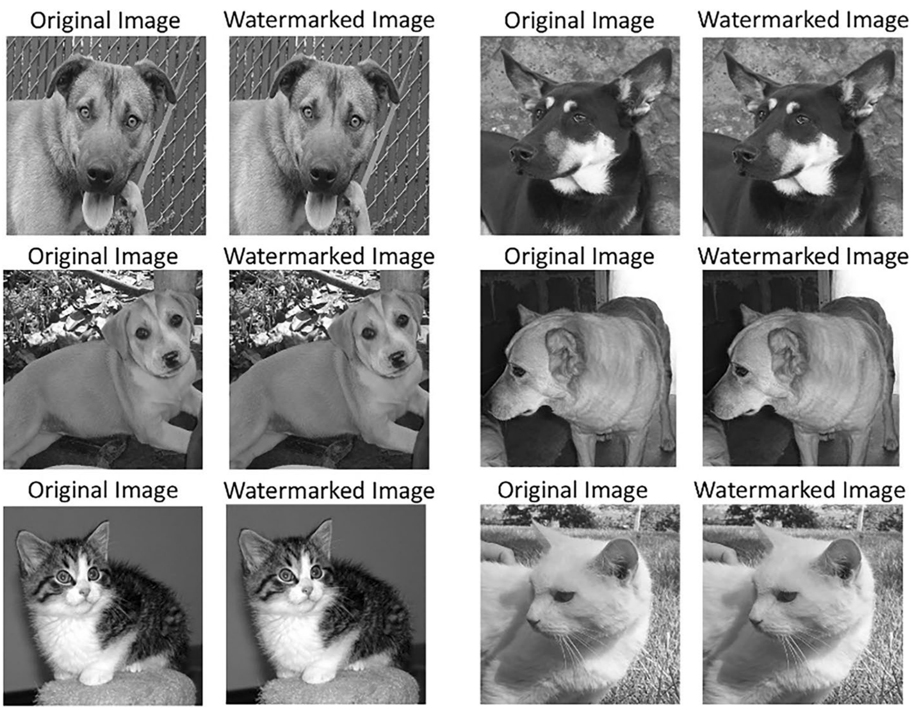


Fig. 7 Test watermarked images from embedding network

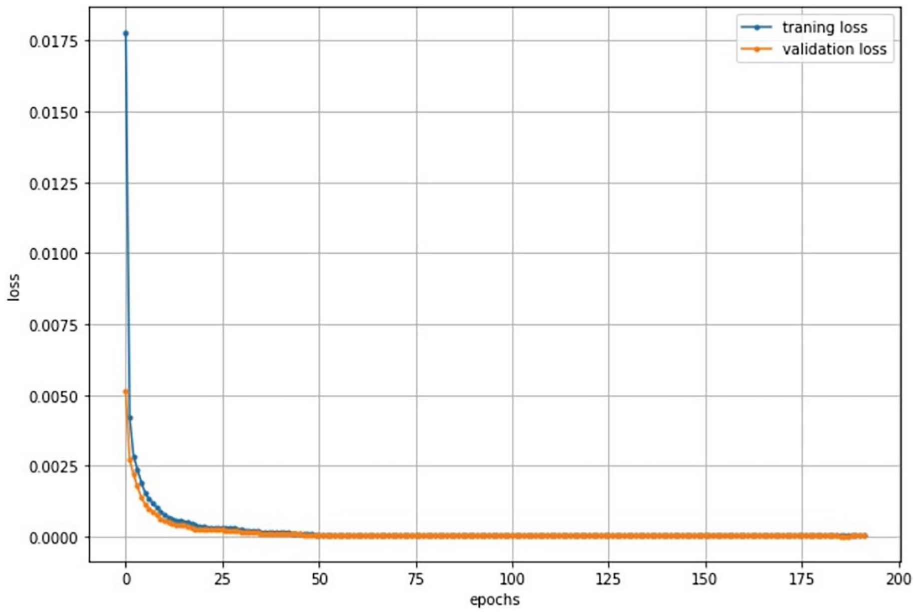


Fig. 8 Training and validation loss for extraction network

4.3 Results and comparison details

In this section, we analyse the effect of image processing attacks on hidden data (mark image) and illustrate the comparison results. Table 3 shows the NC results of the resilience against attacks analysis. From this table, we can note that the NC value is greater than 0.7638, which means that the recovered mark image is acceptable under the considered attacks.

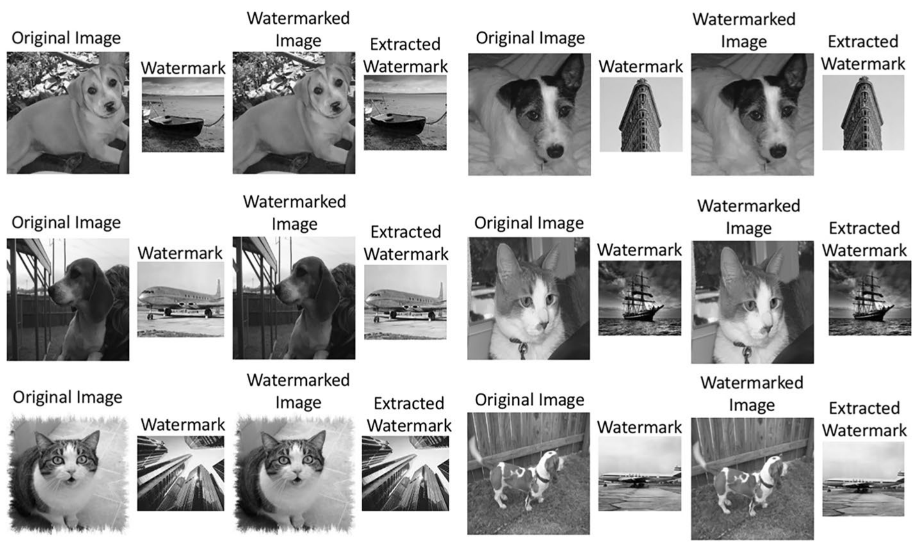


Fig. 9 Test watermark images extracted using the extraction network

Table 3 NC value against general image processing attacks

Gaussian Noise		Salt& Pepper		Speckle		JPEG Compression		Rotation	
Variance	NC	Intensity	NC	Variance	NC	QF	NC	Angle	NC
0.001	0.9634	0.001	0.9912	0.001	0.9942	90	0.8952	1	0.8911
0.003	0.9465	0.004	0.9893	0.003	0.9892	80	0.8916	2	0.8736
0.005	0.9312	0.007	0.9783	0.005	0.9876	70	0.8842	3	0.8582
0.01	0.9172	0.01	0.9702	0.01	0.9783	60	0.8751	4	0.8251
0.03	0.9074	0.03	0.9548	0.03	0.9507	50	0.8604	5	0.7964
0.05	0.8928	0.05	0.9352	0.05	0.9389	40	0.8521	6	0.7638

Table 4 NC values on hybrid attacks

S. No.	Hybrid Attacks	NC Value
1	Gaussian Noise {Variance 0.001} + Salt & Pepper {Intensity 0.001}	0.9523
2	Gaussian Noise {Variance 0.01} + Salt & Pepper {Intensity 0.01}	0.8962
3	Gaussian Noise {Variance 0.001} + Rotation {Angle 1}	0.8596
4	Salt & Pepper {Intensity 0.001} + Rotation {Angle 1}	0.8733
5	Gaussian Noise {Variance 0.001} + JPEG compression {QF 95}	0.9159
6	Salt & Pepper {Intensity 0.001} + JPEG compression {QF 95}	0.9347
7	Speckle {Variance 0.001} + Gaussian Noise {Variance 0.001}	0.9385
8	Speckle {Variance 0.01} + Gaussian Noise {Variance 0.01}	0.8963
9	Speckle {Variance 0.001} + Salt & Pepper {Intensity 0.001}	0.9626
10	Speckle {Variance 0.001} + Salt & Pepper {Intensity 0.001}	0.9247

Table 5 Comparison of PSNR and SSIM with other existing schemes with ours

Methods	Dataset		PSNR (dB)	SSIM	Proposed Scheme	
	Cover Image	Watermark image			PSNR (dB)	SSIM
Wei et al. [23]	CelebA [14]	Random	37.91	0.979	42.872	0.983
Zhong et al. [25]	ImageNet [6]	CIFAR10	39.72	-------	42.583	0.992
Ding et al. [7]	Kaggle	Random	38	0.99	44.6333	0.9996
Mahapatra et al. [16]	Cats & dogs	Random	31.34	0.9940	44.48	0.9997
Rahim et al. [20]	CIFAR10	MNIST	32.9	0.87	44.6814	0.9996
	CIFAR10	CIFAR10	30.9	0.98	44.8923	0.999

Table 6 Comparison of NC against other schemes with ours

Attacks	NC Values				
	Ding et al. [7]	Wang et al. [22]	Zheng et al. [24]	Mahapatra et al. [16]	Proposed
Median filter 3×3	0.1029	0.9906	0.979	0.9877	0.9921
JPEG QF = 95	0.707	0.9624	0.955	0.9501	0.9757
Rotation (45)	0.1496	0.8026	0.7657	0.3895	0.8117

Table 7 Comparison analysis of Mahapatra et al. [16] schemes with ours

Parameters	Mahapatra et al. [16]	Proposed
Number of watermarks	Set of 64 marks	Set of 1000 marks
Number of cover images	6000 for training; 2000 for testing	10,000 for training; 8000 for testing
Image dimension	128×128 (Cover); 64×64 (watermark)	128×128 (Cover); 32×32 & 64×64 (watermark)
Embedder network architecture	Convolution layers for encoder; Transpose convolution for decoder	Autoencoders with feature concatenation
Extractor network architecture	DNN blocks followed by transposed convolution layers	Denoising autoencoders followed by DNN blocks and deconvolutional network
Obtained PSNR	31.34 dB	44.48 dB (32×32) and 41.1 dB (64×64)
NC without attacks	0.9937	0.9996 (32×32) and 0.9982 (64×64)
Embedding and extraction time	9.6 s and 3.19 s	0.4 s and 0.6 Sect. (32×32), 0.7 s and 0.8 Sect. (64×64)

The proposed scheme also showed its advantages by covering more distortion against general image processing attacks and hybrid attacks. The NC values against some hybrid attacks are shown in Table 4. The visual quality performance of the proposed scheme is compared with similar methods [7, 16, 20, 23, 25] in Table 5. We can note that the PSNR and SSIM of our scheme were higher than others. The maximum PSNR and SSIM scores reached 44.8923 dB and 0.999, respectively.

Further, the robustness performance of the proposed scheme is compared with similar methods [7, 16, 22, 24] in Table 6. We can note that the NC score of our scheme, which reached 0.9921, was higher than others. This indicates that the extracted mark image and the original mark are almost the same in their content. Furthermore, we compared our scheme with the scheme of Mahapatra et al. [16] in Table 7. The scheme was compared and analysed based on the network configuration parameters and the results obtained on similar datasets.

5 Conclusion

In this work, CNN-based robust watermarking for digital images is presented. The proposed scheme utilises the learning ability of a deep learning network to automatically learn and generalise the watermarking algorithms and trains it in an unsupervised manner to reduce human intervention. The employment of the embedding and extractor networks ensures that the proposed scheme is imperceptible and protects the mark image satisfactorily against attacks. In conclusion, the proposed technique not only ensures high invisibility and robustness but also improves the performance significantly by up to 41.04% in robustness and 31.1% in invisibility compared with other methods. However, we should improve the embedding capacity in near future for many practical applications. Since dual watermarking contains more authentications and demanding for practical applications, we will report our findings on such watermarking in a future publication. We will further investigate the performance of our algorithm for colour images with improved capacity in our future work.

Acknowledgements This research was supported by project no. DLRL/21CR0003/ SWCC&ENT/GN/LP dt. 29 August, 2020, DLRL, Hyderabad, India.

Declarations

Conflict of interest The authors of this manuscript declare no conflicts of interest.

References

1. Amrit P, Singh AK (2022) Survey on watermarking methods in the artificial intelligence domain and beyond. Comput Commun 188:52–65
2. Anand A, Singh AK, Zhou H (2023) A survey of medical image watermarking: state-of-the-art and research directions. Med Inform Process Secur: Tech Appl 14:325–360. https://doi.org/10.1049/PBHE044E_ch14
3. Anand A, Kumar Singh A (2022) A comprehensive study of deep learning-based covert communication. ACM Trans Multimedia Comput Commun Appl (TOMM) 18(2s):1–19

4. Bagheri M, Mohrekesh M, Karimi N, Samavi S, Shirani S, Khadivi P (2020) Image watermarking with region of interest determination using deep neural networks. In 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, pp 1067–1072
5. Chen J, Zhang J, Debattista K, Han J (2023) Semi-supervised unpaired medical image segmentation through task-affinity consistency. IEEE Trans Med Imaging 42(3):594–605
6. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition. IEEE, pp 248–255
7. Ding W, Ming Y, Cao Z, Lin CT (2021) A generalized deep neural network approach for digital watermarking analysis. IEEE Trans Emerg Top Comput Intell 6(3):613–627
8. Fkirin A, Attiya G, El-Sayed A, Shouman MA (2022) Copyright protection of deep neural network models using digital watermarking: a comparative study. Multimedia Tools Appl 81(11):15961–15975
9. Ge S, Xia Z, Fei J, Sun X, Weng J (2022) A robust document image watermarking scheme using deep neural network. arXiv preprint arXiv:2202.13067
10. Islam M, Roy A, Laskar RH (2018) Neural network based robust image watermarking technique in LWT domain. J Intell Fuzzy Syst 34(3):1691–1700
11. Kaggle Cats vs Dogs dataset. Available at https://www.kaggle.com/c/dogs-vs-cats. Accessed 10 Jan 2023
12. Krizhevsky A, Hinton G (2009) Learning multiple layers of features from tiny images, Technical Report TR-2009, University of Toronto, Toronto
13. Kumar C, Singh AK, Kumar P (2018) A recent survey on image watermarking techniques and its application in e-governance. Multimedia Tools Appl 77:3597–3622
14. Liu Z, Luo P, Wang X, Tang X (2018) Large-scale celebfaces attributes (celeba) dataset. Retrieved August, 15(2018):11
15. Liu Y, Zhang D, Zhang Q, Han J (2022) Part-object relational visual saliency. IEEE Trans Pattern Anal Mach Intell 44(7):3688–3704
16. Mahapatra D, Amrit P, Singh OP, Singh AK, Agrawal AK (2022) Autoencoder convolutional neural network-based embedding and extraction model for image watermarking. J Electron Imaging 32(2):021604
17. Mikołajczyk A, Grochowski M (2018) Data augmentation for improving deep learning in image classification problem. 2018 international interdisciplinary PhD workshop (IIPhDW). IEEE, pp 117–122
18. Mohanty SP, Sengupta A, Guturu P, Kougianos E (2017) Everything You want to know about Watermarking: from paper marks to hardware protection. IEEE Consum Electron Mag 6(3):83–91
19. Panchikkil S, Vegesana SP, Manikandan VM, Donta PK, Maddikunta PKR, Gadekallu TR (2023) An ensemble learning approach for reversible data hiding in encrypted images with fibonacci transform. Electronics 12(2):450
20. Rahim R, Nadeem S (2018) End-to-end trained CNN encoder-decoder networks for image steganography. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, pp 0–0
21. Singh HK, Singh AK (2023) Comprehensive review of watermarking techniques in deep-learning environments. J Electron Imaging 32(3):1–23
22. Wang X, Ma D, Hu K, Hu J, Du L (2021) Mapping based residual convolution neural network for non-embedding and blind image watermarking. J Inform Secur Appl 59:102820
23. Wei Q, Wang H, Zhang G (2020) A robust image watermarking approach using cycle variational autoencoder. Secur Commun Netw 2020:1–9
24. Zheng, W., Mo, S., Jin, X., Qu, Y., Deng, F., Shuai, J., … Long, S. (2018). Robust and high-capacity watermarking for image based on DWT-SVD and CNN. In: 2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA). IEEE, pp 1233–1237
25. Zhong X, Huang PC, Mastorakis S, Shih FY (2020) An automated and robust image watermarking scheme based on deep neural networks. IEEE Trans Multimedia 23:1951–1961