



Article

Image Watermarking Based Data Hiding by Discrete Wavelet Transform Quantization Model with Convolutional Generative Adversarial Architectures

C. Annadurai ¹, I. Nelson ¹, K. Nirmala Devi ², R. Manikandan ^{3,*}  and Amir H. Gandomi ^{4,5,*} 

¹ Department of ECE, Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam, Chennai 603110, Tamilnadu, India

² Department of CSE, Kongu Engineering College, Perundurai, Erode 638060, Tamilnadu, India

³ School of Computing, SASTRA Deemed University, Thanjavur 613401, Tamilnadu, India

⁴ Data Science Institute, Faculty of Engineering and Information Systems, University of Technology Sydney, Ultimo, NSW 2007, Australia

⁵ University Research and Innovation Center (EKIK), Óbuda University, 1034 Budapest, Hungary

* Correspondence: srmanimt75@gmail.com (R.M.); gandomi@uts.edu.au (A.H.G.)

Abstract: Traditional watermarking methods can remove a watermark from an image, making it possible to see the copyright information about the image owner or to estimate similarities using techniques such as bit error rate and normalized correlation. Deep learning is another examination field in AI, and is utilized to develop a deep network to extract objective elements and afterwards distinguish the general environment. To assure the robustness and security of computerized image watermarking, we propose a novel algorithm using convolutional generative adversarial neural networks. This research proposed a novel technique in digital watermarking, with data hiding based on segmentation and classification, using deep learning techniques. The used input images are medical images, including Magnetic Resonance Images (MRI) and Computed Tomography (CT) images, which have been processed for noise removal, smoothing and normalization. The processed image has been watermarked using the Singular Value Decomposition-based discrete wavelet transform quantization model, being segmented and classified using convolutional generative adversarial neural networks. The experimental analysis has been carried out in terms of bit error rate, Structural Similarity Index Measure (SSIM), Normalized Cross-Correlation (NCC), training accuracy, and validation accuracy. This achieved an attained bit error rate of 71%, an SSIM of 56%, a Normalized Cross-Correlation of 71%, a training accuracy of 98%, and a validation accuracy of 95%.

Keywords: digital watermarking; data hiding; segmentation; classification; deep learning



Citation: Annadurai, C.; Nelson, I.; Devi, K.N.; Manikandan, R.; Gandomi, A.H. Image Watermarking Based Data Hiding by Discrete Wavelet Transform Quantization Model with Convolutional Generative Adversarial Architectures. *Appl. Sci.* **2023**, *13*, 804. <https://doi.org/10.3390/app13020804>

Academic Editors: David Megias, Wojciech Mazurczyk and Minoru Kuribayashi

Received: 11 November 2022

Revised: 29 December 2022

Accepted: 30 December 2022

Published: 6 January 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The method of surreptitiously inserting and extracting information from a carrier image is known as digital image watermarking. To make a marked image, the data (i.e., watermark) is hidden within a cover image. However, accurate watermark information extraction is only possible for approved recipients. The watermark can take on a variety of shapes depending on the needs of the user, such as some concealed messages for clandestine communication, some random bits for image protection and verification, or electronic signatures [1]. The watermark can be encoded for a variety of reasons, which include the enhancement of perceivable randomness for further security, by using encryption techniques or reversing the effect of noise using error correction codes for watermark integrity under attacks. While a steganographic system focuses primarily on being undetectable by computer analysis as well as human vision, an image watermarking method frequently prioritizes robustness. Therefore, the watermark should still be visible even if the marked image is corrupted or distorted [2]. An ideal picture watermarking system maintains the watermark under a specified class of distortions, without the need for additional methods.

The resilient picture watermarking systems, however, frequently extract the watermark roughly under harmful attacks, using a variety of encoding techniques for restoration [3]. Generally speaking, the technique of grouping pixels or patches so that each one matches the object class of the group is known as semantic segmentation of images. The semantic segmentation findings are also inpainting for authorized users and hidden for unauthorized users. Since the 2000s, inpainting tasks for various computer vision applications have been a topic of study. To produce patches, a variety of algorithms based on contouring, as well as texture propagation or texture synthesis, have been presented [4]. Convolutional Neural Networks (CNNs) have recently been offered as a way to combine image structure with background data. The work of inpainting is not essential for the inverse problem. For this to work, a hidden watermark is created as a set of coordinates for vertices of the polygon that surround the object of interest, as well as a caption for this object in order to automate the operation.

A compilation of photographs from various spectral bands that together provide a high-resolution grayscale image in near-infrared (NIR) light range is known as a multispectral satellite image. If real objects have discernible boundaries, boundary information can be easily extracted from NIR images. Recognizing the texture of natural objects like fields, forests, lakes, rivers, etc., is, in fact, a more challenging problem [5]. This can be processed using texture analysis, which has seen great advancements since the 1960s and has gone through numerous stages, from statistical to deep learning techniques. Any of the existing approaches can be the best option in a specific situation since texture analysis has too many different functions.

When looking at medical applications, the afterimage cannot be recognized by medical professionals as a sign of the deformation brought on by watermark insertion, since it can affect their judgement. As a result, several academics have concentrated on creating lossless watermarking methods for the medical industry. The extraction of the right features from photos is the key to lossless medical watermarking approaches. A verification image that can be used to confirm the image is created using these image attributes and watermarks. This picture is known as the Ownership Share Picture (OSI). The same image attributes are retrieved for image verification, and the Master Share Image (MSI) can be constructed using this feature [6].

The contribution of this research is as follows:

1. A novel technique in digital watermarking with data hiding, based on segmentation and classification by using deep learning techniques.
2. A method of watermarking using the Singular Value Decomposition-based discrete wavelet transform quantization model.
3. Segmentation and classification of watermarked data by using convolutional generative adversarial neural networks.

The organization of this article is as follows: Section 2 discusses related works for existing watermarking techniques, and Section 3 explains the proposed technique in image watermarking, based on segmentation and classification by deep learning techniques. Section 4 shows the experimental analysis, and Section 5 concludes the research with the future scope.

2. Related Works

Blind and non-blind watermarking methods fall into two main groups. While blind approaches, such as the ones used in one study [7], need the original image in order to extract watermark data, non-blind methods, in fact, need these. Work [8] used mammographic pictures to apply the Fast Gradient Sign Attack (FGSM) assault, by using the "Digital Database for Screening Mammography" (DDSM), which includes both malignant and healthy pictures. While the SSIM index dropped below 0.2, the accuracy declined by more than 30%. Authors in one study [9] used FGSM attack on CT scans and X-rays to find COVID-19. They demonstrated the vulnerability of these models by using VGG16 and InceptionV3, whose accuracies have dropped by up to 90% and 63%, respectively.

Some authors [10] used white-box FGSM and black-box one-pixel techniques to target the NLST dataset. While the one-pixel technique merely decreased the model's accuracy by 2–3%, FGSM decreased it by 36%. The FGSM and PGD attacks were used by some [11] to identify skin cancer on Dermoscopic pictures. The performance of the model dropped by as much as 75%. ResNet50 was used in one study [12] to evaluate several of the most well-known white-box assaults, including FGSM, PGD, C&W, and BIM, on three datasets, with the model's performance dropping by 100% in some circumstances. Others [13] suggested the Adaptive Segmentation Mask (ASMA) technique, which is a focused attack for medical image segmentation. This attack achieves high Intersection-over-Union (IoU) deterioration while producing undetectable samples. By creating adversarial instances, and by using geometrical deformations to represent anatomical as well as intensity fluctuations, researchers [14] presented an attack for medical image segmentation. A wavelet tree quantization was used in one study [15] to offer a blind picture watermarking approach that improved the resistance against geometric attacks, such as rotation, scaling, and cropping. A robust multi-scale full-band picture watermarking based on the Distributed Discrete Wavelet Transform (DDWT) together with the Singular Value Decomposition (SVD) was introduced in one paper [16]. (DDWT). This approach is reasonably resistant to both cropping and rotational attacks. A strong picture watermarking system based on a computer-generated hologram was presented in one study [17], to fend off geometric attacks such as translation, rotation, cropping, flipping, and scaling attacks. By using histogram modification [18], researchers developed an image watermarking method that is resistant to geometrical attacks, such as rotation, cropping, scaling, and translation attacks.

3. System Model

This section discusses a novel technique in digital watermarking with data hiding based on segmentation and classification by deep learning techniques. Here, the used input images are Magnetic Resonance Images (MRI) and Computed Tomography (CT) images, which have been processed for noise removal, smoothening and normalization. The processed image has been watermarked using the Singular Value Decomposition-based discrete wavelet transform quantization model. Regarding watermarked data, it was segmented and classified using convolutional generative adversarial neural networks. The proposed architecture is shown in Figure 1.

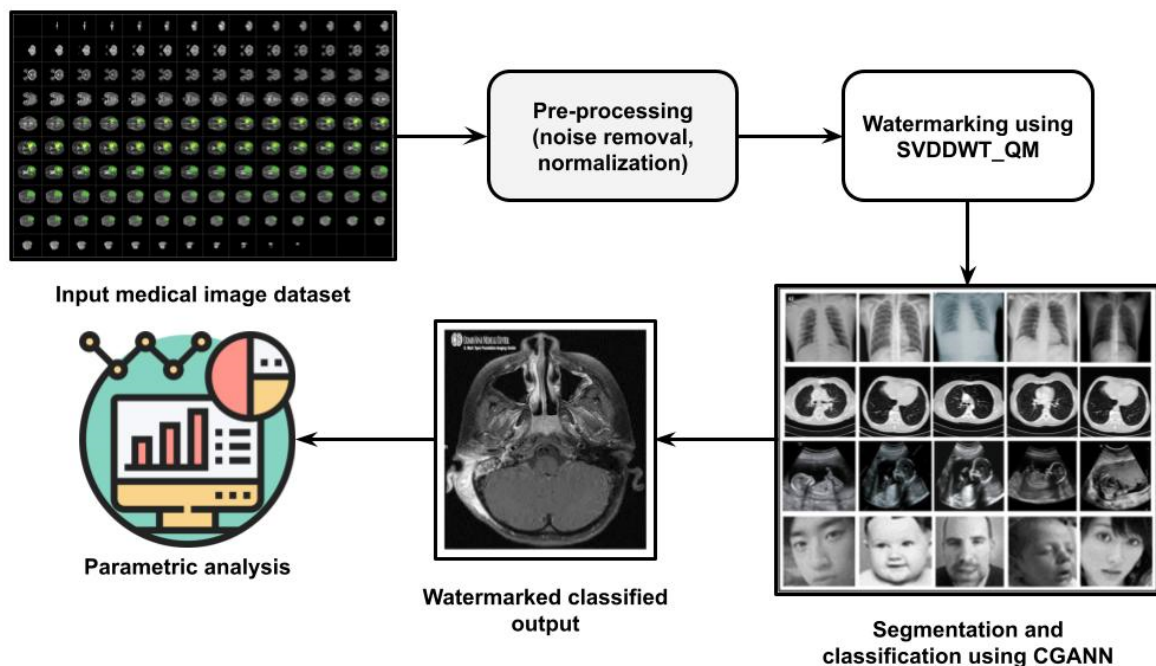


Figure 1. Proposed architecture.

The first step is to reduce the size of the original image from 512×512 pixels to 128×128 pixels, to reduce computation complexity and to assist the network in performing better in less time with simpler calculations. For the system to be trained on unsorted data, as well as to prevent a focus on a certain area of the entire dataset, data were then separated and afterwards shuffled. Data were divided into three sets (training, validation, and test), each one with a separate target label. In order to prevent overfitting and to improve model resilience, images were enhanced with Study I, so the system could recognize them as being brand-new. The photos also received a grayscale distortion, in addition to geometric enhancement.

The watermark was inserted into the recurrence area of the picture. Various images might have unique frequency appropriations. The weaknesses of the installed watermark were changed and consequently founded on the recurrence attributes of the image and, basically, an experimentally foreordained planning capability. The watermark-embedding process was executed in the SVD_DWT area in light of the fact that the SVD_DWT can de-create an image into various frequency parts (or different recurrence subbands). Different frequency parts have various aversions to picture pressure, making it much more straightforward to control the watermark weakness. The weakness of a watermark is chiefly impacted by two factors: how many watermark bits are inserted into every frequency part of the image, and the comparing watermark implanting strength, which is constrained by the quantization boundary. The picture quality was assessed in light of the degradation of the extracted watermark.

Singular Value Decomposition-Based Discrete Wavelet Transform Quantization Model in Image Watermarking:

The most widely used matrix factorization technique in recommendation systems is SVD. In order to capture the linear properties of lncRNAs and illnesses, the SVD technique was applied to the proposed model. SVD specifications are as follows:

In the event that $R \in Rm \times n$ is an association matrix, then the SVD of matrix R is the factorization of 3 matrices ($U, \Sigma,$ and V^T), as shown in Equation (1).

$$R = U \Sigma V^T \tag{1}$$

In Equation (2), the diagonal elements σ_i are referred to as the singular values of the matrix R .

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0 \tag{2}$$

By retaining the k greatest singular values in Equation (3), an approximate representation of the matrix R can be obtained.

$$R \approx U_k \Sigma_k (V_k)^T \tag{3}$$

Output $y^{(j)}$ ($j = 1, 2, 3, \dots, N$) can be described as Equation (4), if there are L hidden nodes and SLFN is a standard one.

$$y^{(j)} = \sum_{i=1}^L v_i g_i(x^{(j)}) = \sum_{i=1}^L v_i g(a_i \cdot x^{(j)} + b_i) \tag{4}$$

$a_i = [a_{i1}, a_{i2}, \dots, a_{in}]$ The weight vector $T_i = 1, 2, \dots, L$) connects the i th hidden node to all input nodes, and $v_i = [v_{i1}, v_{i2}, \dots, v_{im}]$. The weight vector $T_i = 1, 2, 3, \dots, L$) connects the i th hidden node with all output nodes. Equation (5) provides its general expression.

$$HV = Y \tag{5}$$

By choosing the best combinations of $\{V^*, a^*i, b^*i, i = 1, 2, \dots, L\}$, the discrepancy between forecasts and targets can be reduced, as shown in Equation (6).

$$\| \mathbf{H}(a^*, \dots, a^*, b^*, \dots, b^*) \mathbf{V}^* - \mathbf{Y} \| = \min(\| \mathbf{H}(a_1, \dots, a_L, b_1, \dots, b_L) \mathbf{V} - \mathbf{Y} \|)_{ai, bi, \mathbf{V}} \quad (6)$$

Following gradient-descent-based algorithms, the minimization process modifies weights, as well as biases, through iterations of backward propagation. The equation for this method is represented in Equation (7).

$$W_k = W_{k-1} - \eta \frac{\delta E(W)}{\delta W} \quad (7)$$

where E is the error that remains, after each prediction iteration, with the learning rate η . For the vast majority of problems, these gradient-descent-based techniques perform admirably. These algorithms are typically slow since they involve iterative learning stages. Additionally, they have issues with overfitting, which is frequently used to find local minima as opposed to global ones. By increasing the likelihood of visible units, which is represented by the joint probability distribution of visible-hidden unit pair in Equation (8), the ideal set of parameters can be determined.

$$P(v, h) = \frac{e^{-E(v, h)}}{\sum_{v'} \sum_{h'} e^{-E(v', h')}} \quad (8)$$

which has all the possible pairings of visible hidden units as its denominator. The probability of a visible unit is in Equation (9), after marginalizing the space of concealed units.

$$P(v) = \sum_h P(v, h) = \frac{\sum_h e^{-E(v, h)}}{\sum_{v'} \sum_{h'} e^{-E(v', h')}} \quad (9)$$

Tensor $A \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, each of which is $N_2 \times N_3$ in size. By Equation (10) A is decomposed according to Tucker

$$A = S \times \mathbf{U}^{(1)} \times \mathbf{U}^{(2)} \times \mathbf{U}^{(3)} = \sum_{i_1} \sum_{i_2} \sum_{i_3} s_{i_1 i_2 i_3} \mathbf{U}_{i_1}^{(1)} \circ \mathbf{U}_{i_2}^{(2)} \circ \mathbf{U}_{i_3}^{(3)} \quad (10)$$

All images in the y- and x-directions are based on columns of $\mathbf{U}^{(2)}$ and $\mathbf{U}^{(3)}$, respectively. For a particular i_1 , one large part of the leading $S^{(1)}(i_1, i_2, i_3)_{i_2, i_3}$ is picked As a result, the approximation of picture A_{i1} is represented by $\{ \mathbf{U}^{(2)}_{i_2} \circ \mathbf{U}^{(3)}_{i_3} \}_{i_2, i_3}$, $\{ S^{(1)}(i_1, i_2, i_3) \}_{(i_2, i_3) \in \{2, 3\} \times \{1\}}$. Equation (11) provides the covariance between two vectors.

$$\text{Cov}(x_i^1, x_i^2) = E \left[\left(x_i^1 - \mu_{x_i^1}^1 \right) \left(x_i^2 - \mu_{x_i^2}^2 \right) \right] \quad (11)$$

Additionally, the average of all the pixels is given by Equation (12).

$$\mu_{x_i^1}^1 = \left(\frac{1}{k} \right) \sum x_i^1 \text{ and } \mu_{x_i^2}^2 = \left(\frac{1}{k} \right) \sum x_i^2 \quad (12)$$

It is calculated to produce a diagonal matrix D of eigenvalues and a matrix V of corresponding eigenvectors. The detailed coefficients' m_1 and m_2 are likewise evaluated in a similar manner. Weights for the fusion rule are the average of all these m_1 and m_2 , which are given by Equation (13).

$$m_{1(av)} = \frac{m_1 \left(\mathbf{L}_n^{1,2} \right) + m_1 \left(\mathbf{LH}_n^{1,2} \right) + m_1 \left(\mathbf{HL}_n^{1,2} \right) + m_1 \left(\mathbf{HH}_n^{1,2} \right)}{N}; \quad (13)$$

Regarding the fused image, it is represented by Equation (14).

$$z = m_{1(av)} * \mathbf{IM1} + m_{2(av)} * \mathbf{IM2} \quad (14)$$

With the DWT being used as a dyadic sample with parameters x and y , which depend on powers of $x = 2^j$ and $y = k * 2^j$; when $j, k \in \mathbb{Z}$ are replaced with formula (15), the outcome is waves.

$$mw_{j,k} = \frac{1}{\sqrt{|2^j|}}mw(2^{-j}s - k) \tag{15}$$

DTW is given by Equation (16).

$$d_{j,k} = \int_{-\infty}^{+\infty} r(s)2^{-j/2} * mw * (2^{-j}s - k)dt \tag{16}$$

The wavelet coefficients are represented by the notation d_j, k , where k stands for location and j stands for level. This process is classified as the first level, for example: (A2, D2) and (A3, D3). In the following part, a statistical formula was used after obtaining these features. As a result, each signal has 2184 features as a result of the statistical functions being applied, in addition to the signal’s class.

Watermarking and data hiding away for double images can be arranged by one of the accompanying installing strategies: text line, word, or character moving, fixed partitioning of the image into blocks, limit modifications, adjustment of character features, change of run-length examples, and adjustments of halftone images. Information is implanted in text records by moving lines and words separating just barely (1/150 inch.) For example, a text line can be moved to encode a $_1'$ or down to encode a $_0'$; a word can be moved left to encode a $_1'$ or right to encode $_0'$.

Convolutional generative adversarial neural networks (Co_Ge_Ad_NN)-based segmentation and classification of watermarked image:

The generator (G) and discriminator models are two modules that make up the Generative Adversarial Network (GAN) (D), with the two competing with one another to produce the best network performance. The GAN model composition is shown in Figure 2. The adversary is meant to be perplexed by the generator network, while the discriminator is meant to tell the created datasets apart from the actual datasets. A multilayer perceptron-based discriminant network comes after a multilayer perceptron-based generation network. Real samples, or outputs, of the generator network, are chosen by the discriminator’s input. When the discriminator network determines whether or not the generator’s output is a real sample, it may tell from the gradient which type of sample is more similar to the real sample, and then modify the generating network using this knowledge. The GAN’s +e function is written as Equation (17):

$$\min\max V(D, G) = E_{x \sim P_{\text{das}}(x)} \left[\log D\left(\frac{x}{y}\right) \right] + E_{z \sim p_z(z)} \left[\log \left(1 - D\left(G\left(\frac{z}{y}\right)\right) \right) \right] \tag{17}$$

GAN will, however, experience issues such as instability during the training process. In comparison to the initial GAN, the dimensional vector of class probabilities ($p = p_1, p_2, \dots, p_{k+1}$) is given by Equation (18):

$$p_j = \frac{e^{l_j}}{\sum_{i=1}^{k+1} e^{l_i}}, j \in \{1, 2, \dots, k + 1\} \tag{18}$$

A genuine image will be distinguished from a fraudulent image as one of the former k classes, and vice versa. Equation (19) was used to represent the Co_Ge_Ad_NN loss function, as a typical minimax game.

$$L = -E_{x,y \sim P_{\text{data}}(x,y)} \{D(y | x, y < k + 1)\} - E_{x \sim G(z)} \{D(y | G(z), y = k + 1)\} \tag{19}$$

After selecting the cross-entropy function as the loss function, $D(y | x)$ was calculated as shown in Equation (20):

$$D(y | x) = - \sum_i y'_i \log(p_i) \tag{20}$$

If y_0 denotes the anticipated class, p_i denotes the likelihood that the incoming sample conforms to y_0 . Equation (20) states that when input is a real image, $D(y | x, y_{k+1})$ is further stated as Equation (21):

$$D(y | x, y < k + 1) = - \sum_{i=1}^k y'_i \log(p_i) \tag{21}$$

$D(y | x, y_{k+1})$ is condensed to the following equation when the input is a fictitious image by Equation (22):

$$D(y | x, y = k + 1) = - \log(p_{k+1}) \tag{22}$$

It is assumed that each training iteration contains m inputs for both the discriminator as well as generator, and that the discriminator is updated by ascending its stochastic gradient by Equation (23):

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m [D(y | x^i, y < k + 1) + D(y | G(z^i), y = k + 1)] \tag{23}$$

As the generator is updated, the stochastic gradient is descended using Equation (24):

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m D(y | G(z^i), y = k + 1) \tag{24}$$

The network of both the generator and the discriminator are optimized while alternately updating them. As a result, both the discriminator and the generator are able to distinguish the input sample from the output sample with greater accuracy. DFNN is called a CNN feature extraction by layer-by-layer learning input image. The batch norm layer is almost always used in the generator and discriminator to normalize the output layers of features, which speeds up training, as well as increases stability. Additionally, the discriminator uses a leaky ReLU activation function to avoid gradient sparseness. The generator’s network structure diagram is represented in Figure 2.

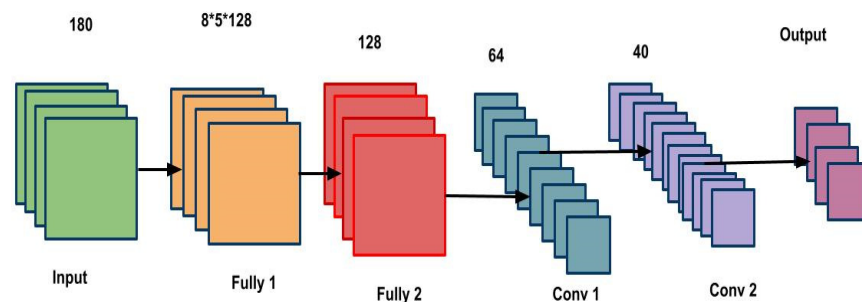


Figure 2. Network structure of the generator of Co_Ge_Ad_NN.

In the generator network shown in Figure 3, each input is combined with a random input (noise produced with Gaussian distribution) to muddle the original image and create a new image, with all the input photos being subjected to this. The generator also engages in up-sampling, which brings together a larger collection of smaller images to create a single enormous image. There are two hidden layers in this system. In order to ensure that neuron activation functions do not take place in zero or dead regions, weights are initialised using the Xavier initialiser. By doing batch normalization in each layer for standardisation, the number of computation-intensive epochs also decreases.

The generator structure is seen in reverse in Figure 4. The discriminator performs downsampling, which means that it reduces the size of the huge image that was generated as a result of upsampling. To identify whether the created image is real or fake, the

discriminator has two hidden layers and uses the “Sigmoid function” as the activation function in the output layer.

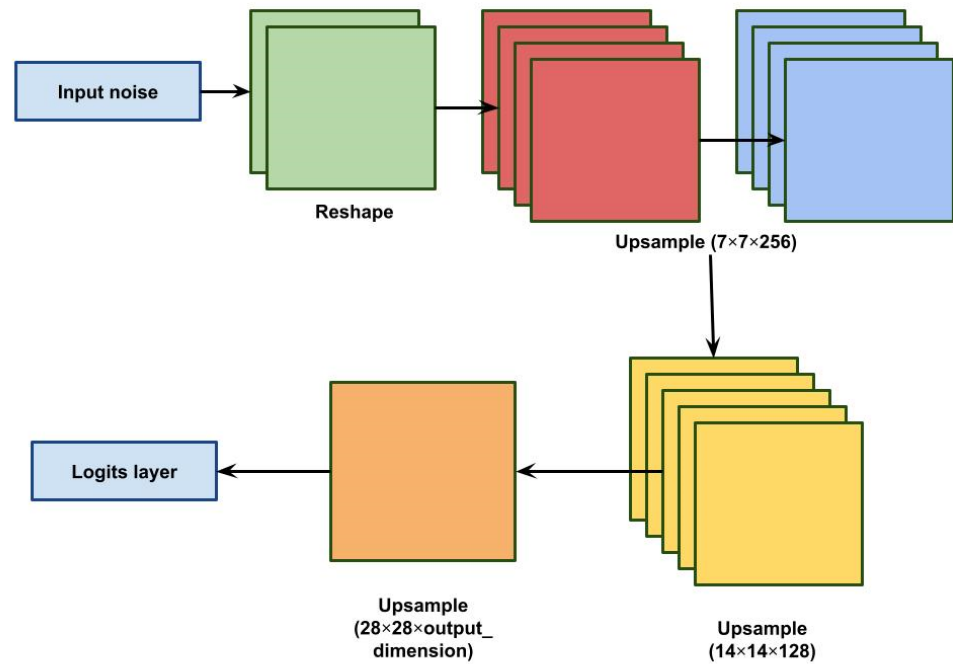


Figure 3. The Generator Network.

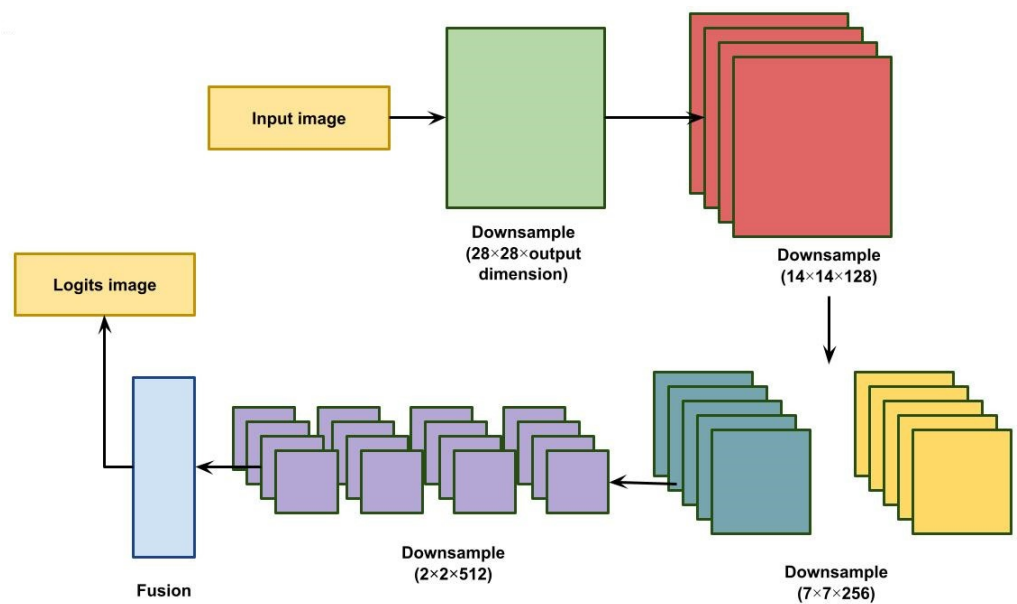


Figure 4. The Discriminator Network.

Convolutional layers, pooling layers, and fully connected layers make up the CNN’s three-layer structure, which was employed to extract features. Different layers serve various purposes. Each of the several neurons that make up these functional layers contacts only a portion of the neurons in the layer above. This enhances the calculation performance, lowering the network’s complexity. The convolutional layer, which is regarded as a feature extraction layer, is made up of a few convolutional neurons. The size of the resulting feature map depends on the size of the convolution kernel. Equation (25) was used to find the size of the feature map after the kernel function was been convolved as well as transported:

$$\begin{cases} N_x^l = \frac{N_x^{l-1} - K_x^l + 2P_x^l}{S_x} \\ N_y^l = \frac{N_y^{l-1} - K_y^l + 2P_y^l}{S_y} \end{cases} \quad (25)$$

where P is the fill pixel value, K is the convolution kernel size, S is the step size, and l is the current number of layers. It is a nonlinear change of activation function following the convolution procedure. Each neuron’s weights and parameters were obtained via the backpropagation technique, and the convolutional layer neuron’s expression is given by Equation (26).

$$x_j^l = \text{Relu} \left(\sum_{i \in \mathcal{M}_j} x_i^{l-1} w_{ij}^l + b_j^l \right) \quad (26)$$

where w and b stand for the connection weight and offset, respectively, and M is the filter size. Feature mapping is carried out via the pooling layer, typically between two convolutional layers. There are two types of pooling processes: maximum and average. Overlapping application pools are also suggested in [16]. The values of particular features in the input layer were evaluated as well as merged in order to decrease the number of neurons, while maintaining the features’ integrity. This was done by minimizing the variance of transformed data through the subsampling layer. The pooling layer’s equation is given in Equation (27):

$$y = \max(x_i), x_i \in x \quad (27)$$

where xi is the output of the neuron in the region denoted by the letter x on the feature map. All the neurons from the previous convolutional layer are linked to the current layer via a fully connected layer, which transforms all local features into global features. The neural network portion of the proposed network has three completely connected layers. Overfitting issues are more likely to occur in fully coupled layers. A dropout function was employed to lessen the overfitting of the first two layers, and in order to solve this issue. The softmax function gives the probability as Equation (28).

$$y_j = \frac{\exp(f_j)}{\sum_{i=1}^2 \exp(f_i)}, j = 1, 2, \quad (28)$$

where yj is the jth neuron’s output probability. The framework for Co_Ge_Ad_NN is shown in Figure 5. To create smoke images with different shapes and textures, smoke images from datasets were fed into the Co_Ge_Ad_NN. Additionally, extra photos that might lead to erroneous detection were also produced.

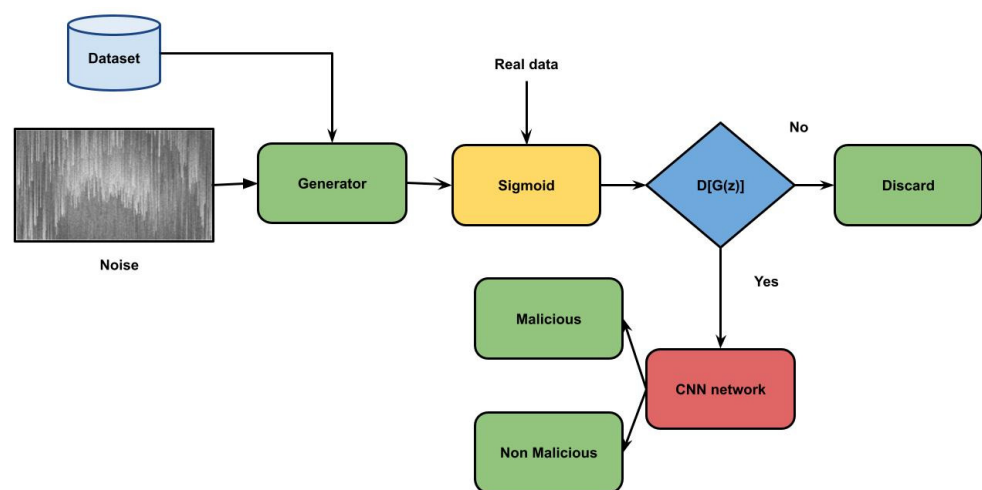


Figure 5. Co_Ge_Ad_NN framework.

All layers must be frozen because the discriminator cannot adjust the parameters during training. Numerous discriminators were trained and several train generators, so the network training can be sped up, which can also increase the network's overall training rate. Additionally, various epochs were set, so the produced smoke images could be analyzed after a number of epochs, and so it was possible to choose the best smoke images. The CNN model was optimised using stochastic gradient descent (SGD). The model learning rate was considered as 0.01 in the experiment since the learning rate can affect the convergence rate. The network model weights were updated by SGD, by combining the gradient, as well as by the modified weight from the previous iteration; the entire procedure is described by the two following Equations (29) and (30):

$$V_{t+1} = \mu V_t + \alpha \nabla L(W_t) \quad (29)$$

$$W_{t+1} = W_t - V_{t+1} \quad (30)$$

where W_{t+1} denotes the weight of the network after $t+1$ iterations, and V_{t+1} denotes the weight of the network after $t+1$ iterative updates. An appropriate loss function is essential to the network's performance throughout the network model's training. Currently, common loss functions include hinge, log, and contrastive losses. Because it is better suited for binary classification issues, we opted to use the cross-entropy loss function. The cross-entropy loss function is given by Equation (31).

$$H(p, q) = - \sum_x p(x) \log q(x) \quad (31)$$

The input and output labels are represented by $p(x)$ and $q(x)$, respectively. To show the value of hyperparameters in network design, several tests were run, showing that the proposed network is greatly impacted by the configuration of many hyperparameters. In conclusion, overlapping max pooling outperforms nonoverlapped max pooling layers in terms of performance. Additionally, properly lowering the number of neurons in completely linked layers not only accelerates convergence but also enhances detection recognition.

4. Performance Analysis

All the trials were conducted by using the GforceGTX 770 and CUDNN 5110 graphics cards, together with Python 2.7 and the Spyder Integrated Development Environment (IDE). Using a standard dataset and a learning rate of 0.5, the performance of the proposed watermarking framework was evaluated. Various epochs were used to train our model.

Dataset description: In this study, two separate datasets were used. The first one was obtained between 2005 and 2010 from Nanfang Hospital and General Hospital at Tianjin Medical University in China. It was then published online in a number of editions beginning in 2015 and ending with its final release in 2017. Axial, coronal, and sagittal images are all included in the dataset. The collection contains T1-weighted contrast-enhanced pictures from 233 patients with meningioma, glioma, and pituitary tumours, which are three different forms of brain cancers. Depending on their types and grades, brain tumours can vary in size, location, and shape.

The Cancer Imaging Archive (TCIA), a public access resource, provided a second dataset. In this dataset, there are 130 patients with various diseases, grades, races, and ages, whose MRI multi-sequence pictures can be found in the Repository of Molecular Brain Neoplasia Data (REMBRANDT). T1-weighted contrast-enhanced images that contained gliomas of various grades were selected.

Additionally, the Free Mammogram Database patient data were searched for mammograms (MIAS). Ten mammograms were additionally used to train the classifier. The 46 mammograms comprise 10 benign and 36 malignant masses that are in thick glandular tissue locations, ducts, breast boundaries, blood vessels, and/or glandular tissues in various breast places.

Table 1 shows a comparative analysis between the proposed and existing techniques based on various medical images. Here, the used datasets are TCIA, REMBRANDT, and MIAS datasets. The parameters analysed are bit error rate, SSIM, NCC, training accuracy, and validation accuracy. Here the training and testing have been carried out based on a number of epochs.

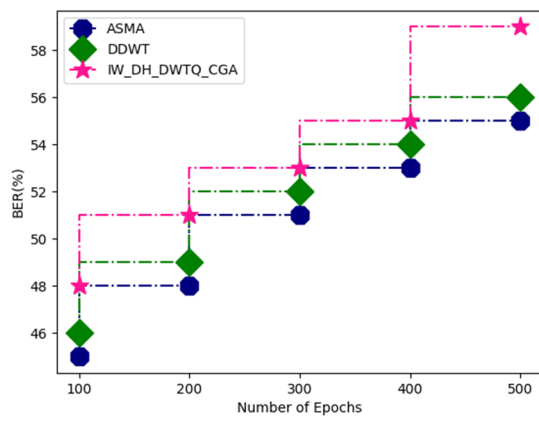
Table 1. Comparative analysis between proposed and existing techniques based on various medical images.

Dataset	Techniques	BER	SSIM	NCC	Training Accuracy	Validation Accuracy
TCIA	ASMA	55	42	52	85	81
	DDWT	56	43	54	88	83
	IW_DH_DWTQ_CGA	59	45	56	89	85
REMBRANDT	ASMA	61	51	58	92	82
	DDWT	63	53	61	93	85
	IW_DH_DWTQ_CGA	65	55	63	95	89
MIAS	ASMA	67	52	65	94	91
	DDWT	68	54	68	96	93
	IW_DH_DWTQ_CGA	71	56	71	98	95

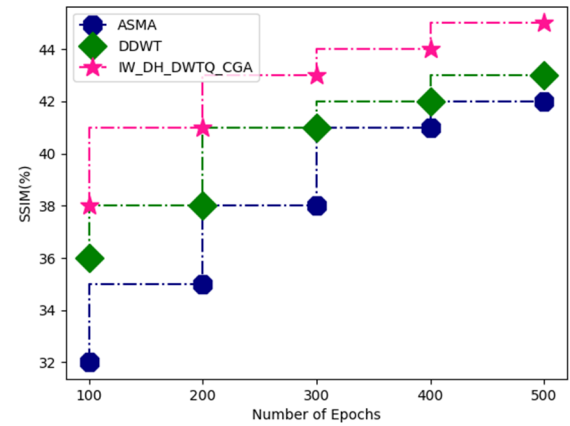
The above Figure 6a–e shows a comparative analysis between the proposed and existing techniques for the TCIA dataset. The proposed technique attained a bit error rate of 59%, an SSIM of 45%, an NCC of 56%, a training accuracy of 89%, and a validation accuracy of 85%. ASMA attained a bit error rate of 55%, an SSIM of 42%, an NCC of 52%, a training accuracy of 85%, and a validation accuracy of 81%. Regarding DDWT, it attained a bit error rate of 56%, an SSIM of 43%, a Normalized Cross-Correlation of 54%, a training accuracy of 88%, and a validation accuracy of 83%.

Figure 7a–e shows a comparative analysis between the proposed and existing technique for the REMBRANDT dataset. The proposed technique attained a bit error rate of 65%, an SSIM of 55%, a Normalized Cross-Correlation of 63%, a training accuracy of 95%, and a validation accuracy of 89%. Additionally, ASMA attained a bit error rate of 61%, an SSIM of 51%, a Normalized Cross-Correlation of 58%, a training accuracy of 92%, and a validation accuracy of 82%. In the case of DDWT, it attained a bit error rate of 63%, an SSIM of 53%, a Normalized Cross-Correlation of 61%, a training accuracy of 93%, and a validation accuracy of 85%.

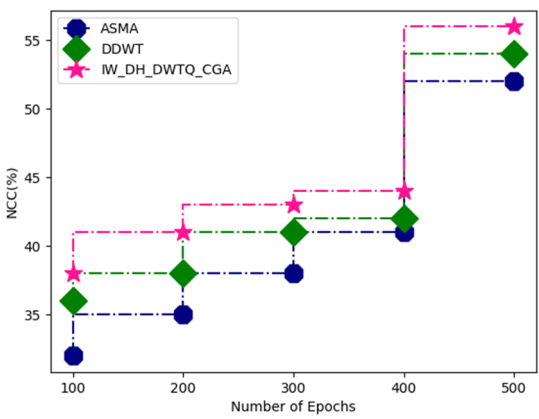
Figure 8a–e shows a comparative analysis between the proposed and existing techniques for the MIAS dataset. The proposed technique attained a bit error rate of 71%, an SSIM of 56%, a Normalized Cross-Correlation of 71%, a training accuracy of 98%, and a validation accuracy of 95%. Regarding ASMA, this attained a bit error rate of 67%, an SSIM of 52%, a Normalized Cross-Correlation of 65%, a training accuracy of 94%, and a validation accuracy of 91%. DDWT attained a bit error rate of 68%, an SSIM of 54%, a Normalized Cross-Correlation of 68%, a training accuracy of 96%, and a validation accuracy of 93%.



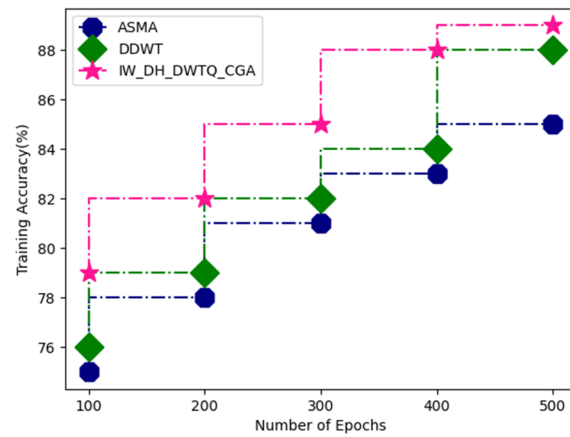
(a) BER



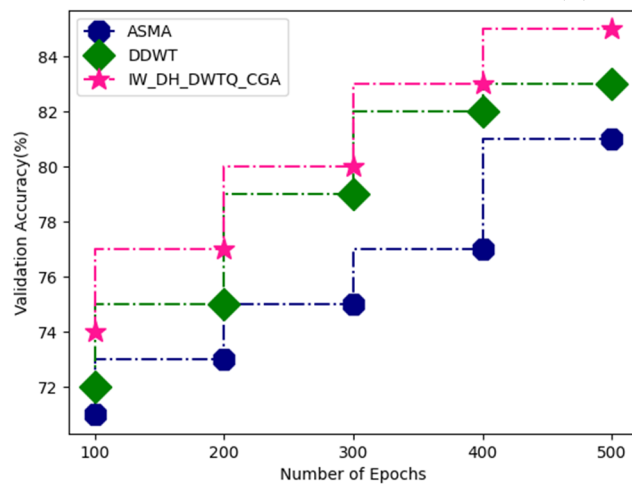
(b) SSIM



(c) NCC

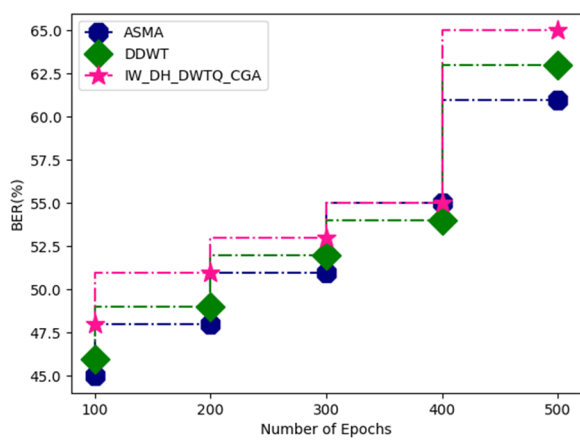


(d) Training accuracy

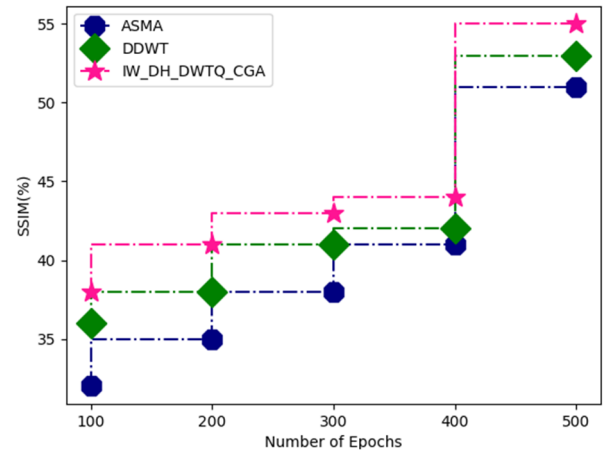


(e) Validation accuracy

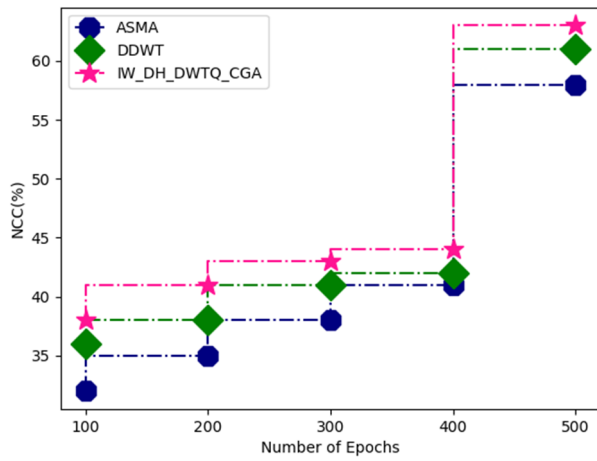
Figure 6. Comparative analysis between proposed and existing technique for TCIA dataset in terms of (a) bit error rate, (b) SSIM, (c) NCC, (d) training accuracy, (e) validation accuracy.



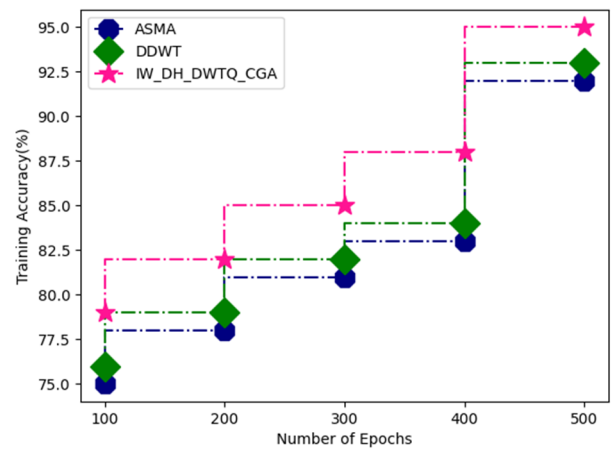
(a) bit error rate



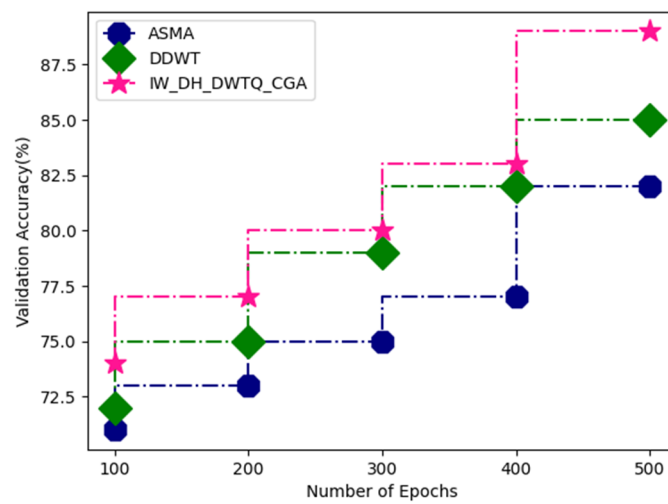
(b) SSIM



(c) NCC

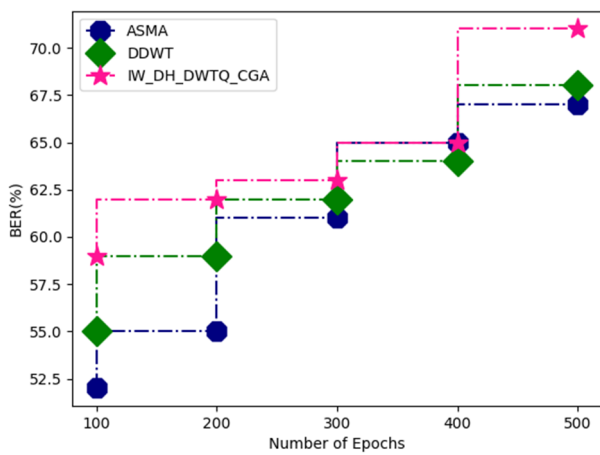


(d) training accuracy

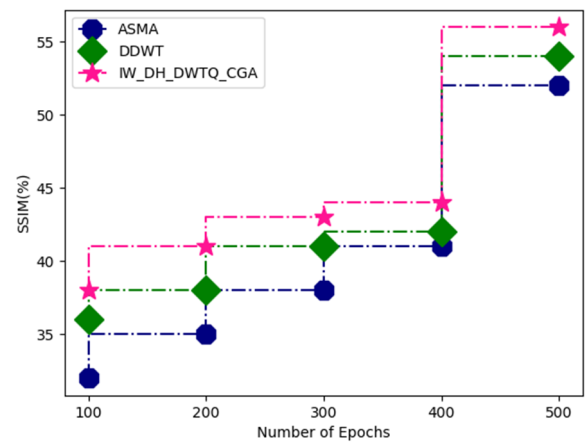


(e) validation accuracy

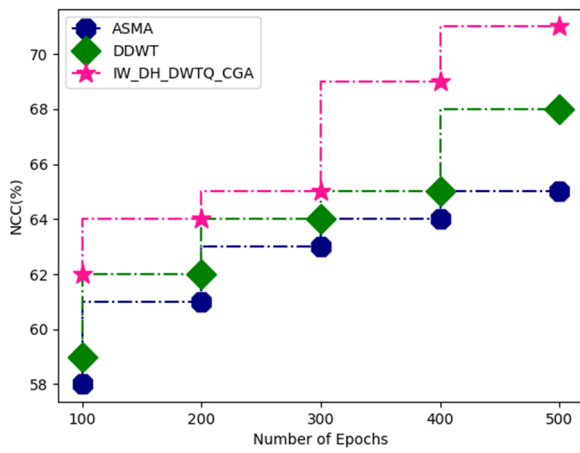
Figure 7. Comparative analysis between proposed and existing techniques for the REMBRANDT dataset in terms of (a) bit error rate, (b) SSIM, (c) NCC, (d) training accuracy, (e) validation accuracy.



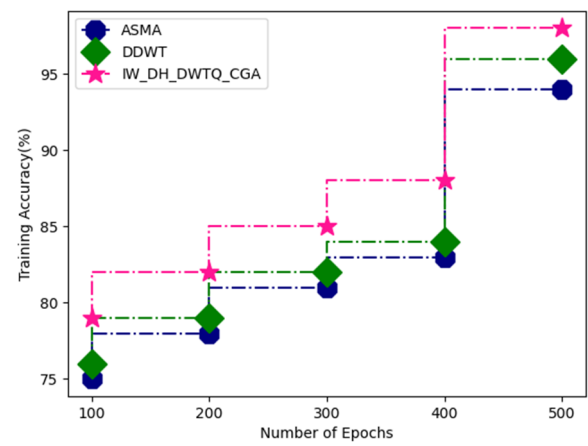
(a) bit error rate



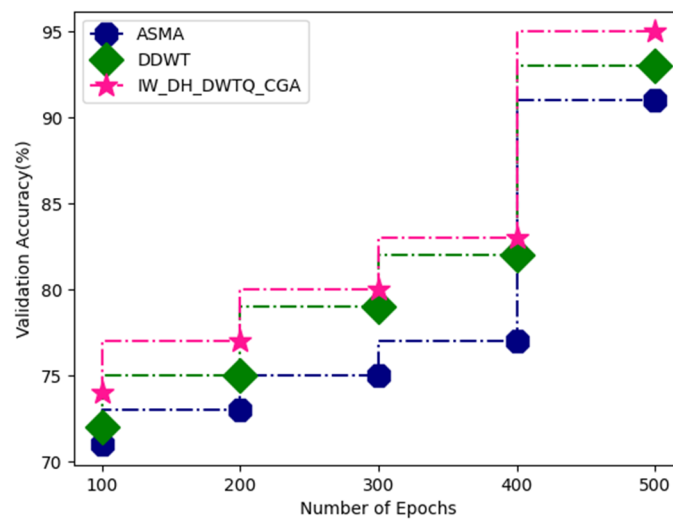
(b) SSIM



(c) NCC



(d) training accuracy



(e) validation accuracy

Figure 8. Comparative analysis between proposed and existing techniques for the MIAS dataset in terms of (a) bit error rate, (b) SSIM, (c) NCC, (d) training accuracy, (e) validation accuracy.

5. Discussion

The watermark is inserted by quantizing the wavelet change coefficients of the image, and the debasement of the watermark mirrors the corruption of image quality. The main part of the proposed strategy is that the image quality assessment of compacted pictures can be accomplished without the requirement of obtaining data relating to the first images. The precision of value assessment is kept up via naturally changing the watermark weakness as per the different recurrence conveyance of each image. The tests show that the proposed watermarking system is robust against image handling attacks and resistant to re-encoding assaults. The watermarking security is forced by arbitrarily choosing the applicant blocks. We directed a progression of tests to demonstrate its viability concerning perceptual quality and expansion in image bit rate.

6. Conclusions

This research proposes a novel technique in digital watermarking, with data hiding based on segmentation and classification by deep learning techniques. The processed image has been watermarked using the SVD-based discrete wavelet transform quantization model. Moreover, the watermarked image was segmented and classified using convolutional generative adversarial neural networks. In order to examine deep learning's intellectual property, embed watermarks, and verify owner information, a deep learning model is suggested for digital watermarking. This model can provide the watermarks needed to defend against potential attacks. The common dataset was used to test the proposed dataset. This achieved an attained bit error rate of 71%, an SSIM of 56%, a Normalized Cross-Correlation of 71%, a training accuracy of 98%, and a validation accuracy of 95%. In future work, the authors plan to experiment with other moment families and watermarking strategies while still adhering to the general framework outlined above. There is also an intention to research other widely used wavelet-based or other popular medical image watermarking methods.

Author Contributions: Conceptualization, C.A., I.N., K.N.D., R.M. and A.H.G.; methodology, C.A., I.N., K.N.D., R.M. and A.H.G.; software, C.A., I.N., K.N.D. and R.M.; validation, C.A., I.N., K.N.D. and R.M.; formal analysis, C.A., I.N., K.N.D. and R.M.; investigation, C.A., I.N., K.N.D., R.M. and A.H.G.; resources, C.A., I.N., K.N.D., R.M., A.H.G.; data curation, C.A., I.N., K.N.D., R.M. and A.H.G.; writing—original draft preparation, C.A., I.N., K.N.D., R.M. and A.H.G.; writing—review and editing, C.A., I.N., K.N.D., R.M. and A.H.G.; visualization, C.A., I.N., K.N.D., R.M. and A.H.G.; supervision, A.H.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Will be available on request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sun, Z.; Chang, N.B.; Chen, C.F.; Mostafiz, C.; Gao, W. Ensemble learning via higher order singular value decomposition for integrating data and classifier fusion in water quality monitoring. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 3345–3360. [[CrossRef](#)]
2. Zeng, M.; Lu, C.; Zhang, F.; Li, Y.; Wu, F.X.; Li, Y.; Li, M. SLDLA: lncRNA-disease association prediction based on singular value decomposition and deep learning. *Methods* **2020**, *179*, 73–80. [[CrossRef](#)] [[PubMed](#)]
3. Vijayarajan, R.; Muttan, S. Discrete wavelet transform based principal component averaging fusion for medical images. *AEU-Int. J. Electron. Commun.* **2015**, *69*, 896–902. [[CrossRef](#)]
4. Yin, H.; Wei, Y.; Liu, H.; Liu, S.; Liu, C.; Gao, Y. Deep convolutional generative adversarial network and convolutional neural network for smoke detection. *Complexity* **2020**, *2020*, 101–113. [[CrossRef](#)]
5. Hatoum, M.W.; Couchot, J.F.; Couturier, R.; Darazi, R. Using deep learning for image watermarking attack. *Signal Process. Image Commun.* **2021**, *90*, 116019. [[CrossRef](#)]

6. Wang, Y.; Li, W.; Zhang, Y. Research Article Mathematical Model Design of the Traditional Dress Recognition Algorithm Based on Digital Watermarking Technology. *Math. Probl. Eng.* **2022**, *2022*, 455–465.
7. Ge, S.; Xia, Z.; Fei, J.; Sun, X.; Weng, J. A Robust Document Image Watermarking Scheme using Deep Neural Network. *arXiv* **2022**, arXiv:2202.13067.
8. Begum, M.; Uddin, M.S. Towards the development of an effective image watermarking system. *Secur. Priv.* **2022**, *5*, e196. [[CrossRef](#)]
9. Hemalatha, B.; Karthik, B.; Balaji, S.; Senthilkumar, K.K.; Ghosh, A. CNN Based Image Forgery Segmentation and Classification for Forensic Verification. In *International Conference on Electrical and Electronics Engineering*; Springer: Singapore, 2022; pp. 652–661.
10. Azam, M.A.; Khan, K.B.; Salahuddin, S.; Rehman, E.; Khan, S.A.; Khan, M.A.; Kadry, S.; Gandomi, A.H. A review on multimodal medical image fusion: Compendious analysis of medical modalities, multimodal databases, fusion techniques and quality metrics. *Comput. Biol. Med.* **2022**, *144*, 105253. [[CrossRef](#)] [[PubMed](#)]
11. Nanammal, V.; Venugopalakrishnan, J.>NNLGBM: Medical Image Classification through Secure Collaboration in Pneumonia Detection by Blending NN and LGBM. *Int. J. Intell. Syst. Appl. Eng.* **2022**, *10*, 201–212.
12. Singh, P.K. Robust and imperceptible image watermarking technique based on SVD, DCT, BEMD and PSO in wavelet domain. *Multimed. Tools Appl.* **2021**, *81*, 22001–22026.
13. Ito, H.; AprilPyone, M.; Kiya, H. Access control using spatially invariant permutation of feature maps for semantic segmentation models. In Proceedings of the 2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Tokyo, Japan, 14–17 December 2021; pp. 1833–1838.
14. Eze, P.; Parampalli, U. Deep Learning Evaluation of a Steganographic Algorithm. In Proceedings of the 2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Tokyo, Japan, 14–17 December 2021; pp. 1999–2005.
15. Wang, X.; Ma, D.; Hu, K.; Hu, J.; Du, L. Mapping based residual convolution neural network for non-embedding and blind image watermarking. *J. Inf. Secur. Appl.* **2021**, *59*, 102820. [[CrossRef](#)]
16. Kumar, S.; Rajpal, A.; Sharma, N.K.; Rajpal, S.; Nayyar, A.; Kumar, N. ROSEmark: Robust semi-blind ECG watermarking scheme using SWT-DCT framework. *Digit. Signal Process.* **2022**, *129*, 103648. [[CrossRef](#)]
17. Yoo, I.; Chang, H.; Luo, X.; Stava, O.; Liu, C.; Milanfar, P.; Yang, F. Deep 3D-to-2D Watermarking: Embedding Messages in 3D Meshes and Extracting Them from 2D Renderings. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 10031–10040.
18. Balliwal, R.; Saini, S.S. Design Simulation of Predicting Age and Gender for Human using Machine Learning Approach. *J. Online Eng. Educ.* **2022**, *13*, 6–16.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.