Deep Learning-Based Data Hiding

Sukanya Dutta, Supratim Maity, Hirak Kumar Maity

College of Engineering & Management, Kolaghat, Purba Medinipur, W.B., India

Corresponding author: Hirak Kumar Maity, Email: hmaity@cemk.ac.in

Data hiding, also known as steganography, is an essential technique for concealing sensitive information within digital media to ensure secure communication and protect data confidentiality. After 1950, in recent years, deep learning methods have revolutionized various domains, including computer vision and natural language processing. This paper introduces a robust watermark embedding technique using CNN (Convolutional Neural Network). By leveraging the power of deep neural networks and using the transform domain technique as DWT (Discrete Wavelet Transform) and also the domain-specific technique as SVD (Singular Value Decomposition), we propose an efficient and robust data-hiding framework that surpasses traditional methods in terms of capacity, security, and imperceptibility. Experimental results demonstrate the effectiveness of our proposed approach in various scenarios, highlighting its potential for real-world applications.

Keywords: Deep learning, Data hiding, Steganography, Convolutional neural networks, Information security, Imperceptibility.

2023. In Satyasai Jagannath Nanda & Rajendra Prasad Yadav (eds.), *Data Science and Intelligent Computing Techniques*, 551–567. Computing & Intelligent Systems, SCRS, India. https://doi.org/10.56155/978-81-955020-2-8-51

1 Introduction

With the advancement of technology, data security is now much more vulnerable. In the current era of advanced technology, the exchange of digital data over the internet has become increasingly accessible [25], [1] and also preventable. For prevention purposes, one of the fundamental approaches to ensuring the integrity and authenticity of digital content is through the use of authentication techniques, such as watermarking. Watermarking involves the embedding of a unique and imperceptible identifier into the digital data, serving as a digital signature to verify its authenticity and integrity. Watermarking allows the recipient to verify the origin of the digital content. By extracting the embedded watermark, the recipient can validate the authenticity of the content and ensure it has not been tampered with during transmission. This extraction process can only be provided by a dedicated detector [2], [3]. Even, in case of copyright infringement, the embedded watermark can provide evidence of ownership, facilitating legal actions and copyright enforcement [26], [4], [5]. Watermarking allows for monitoring the spread and usage of digital content. By embedding different watermarks in copies distributed to different recipients, the source of unauthorized dissemination can be identified, aiding in tracking the unauthorized distribution channels [6], [27]. Deep learning is a field of ML (Machine Learning) but instead of using ML, deep learning is more efficient. ML still uses traditional techniques; Deep learning automatically extracts the features. Often, ML algorithms are unable to find nonlinearity; in that case, deep learning can capture all of those. Professor Geoffrey Hinton proposed the concept of deep learning and improved the training methods for models, overcoming the limitations of traditional backpropagation neural networks. Deep learning, especially convolutional neural networks (CNNs), has since become a focal point of research across various scientific domains [28]. However, deep learning is not always superior to ML in all watermarking scenarios. ML approaches can still be effective in certain cases, especially when the watermarking requirements are simpler, the dataset is small, or interpretability is crucial. In this paper, by integrating deep learning techniques with digital watermarking, we intend to improve the resistance of watermarked content against various attacks and manipulations. This integration has the potential to advance the effectiveness and security of digital watermarking systems [12], [10].

The objective of this research is to explore the application of convolutional neural networks in digital watermarking, leveraging their ability to extract meaningful features from multimedia content. By exploiting the power of deep learning, we aim to develop a robust and efficient data-hiding framework that surpasses traditional methods in terms of image size, security, and imperceptibility. Through extensive experiments and analysis, we evaluate the performance of our proposed approach and demonstrate its potential for real-world applications.

The organization of this paper is as follows: Section 2 provides a brief review of the related work. The proposed architecture is discussed in section 3. Section 4 represents the simulation result, and finally, the conclusion and scope of future work are given in section 5.

2 Related Works

The related works, such as data hiding and deep learning algorithms are mentioned in this section.

2.1 Digital Watermarking Techniques:

Digital watermarking techniques have been extensively studied and developed to address the challenges of protecting digital content from unauthorized use and tampering. Two widely used techniques in this domain are Discrete Wavelet Transform (DWT) and Singular Value Decomposition (SVD).In recent years, these two techniques are mostly used for the purpose of enhancing the robustness and high capacity of multi-size images [11] , [13].

Discrete Wavelet Transform (DWT): DWT is a popular transform domain technique, used for image and signal processing. This technique aimed at ensuring robust watermarking with the utilization of deep learning paradigms [14]. It decomposes an image or signal into different frequency subbands, allowing for efficient analysis and manipulation of the data. In the context of watermarking, DWT can be utilized to embed the watermark into the transform coefficients of the host image. The watermark can be embedded in the subbands, where it is less perceptible to human vision. In this paper, we applied many levels of haar wavelet transform to get a refined approximation (LL), and in every level of decomposition, the HH band gets high-frequency features. This technique gives robustness. DWT-based watermarking techniques offer robustness against common attacks and provide a good trade-off between imperceptibility and robustness [7], [8]. For authenticity and proof purposes, digital watermarking is used

in the medical industry also and the DWT technique proves its robustness effectively [17].

Singular Value Decomposition (SVD): SVD is a mathematical technique that enhances the security and resilience of their steganography technique, ensuring the preservation of hidden information even in the presence of various attacks [18]. SVD decomposes a matrix into three components: a unitary matrix, a diagonal matrix, and the conjugate transpose of the unitary matrix. In image watermarking, SVD can be employed to embed the watermark by modifying the singular values of the host image [29]. By replacing some of the singular values with the watermark information, it is possible to embed the watermark robustly and imperceptibly. SVD-based watermarking techniques are known for their high capacity and resilience against various attacks, including cropping, compression, and noise addition.

2.2 Deep Learning-Based Data Hiding:

Several studies have explored the application of deep learning models, particularly convolutional neural networks (CNNs), for data hiding and information security. In [9], Yedroudj proposes DeepSteg, a deep learning-based framework for both steganalysis and steganography tasks. They employ CNN architectures to detect the presence of hidden information in images and also to embed secret information within cover images. The model is trained on a large-scale dataset and achieves high detection accuracy and imperceptibility. The authors demonstrate the effectiveness of DeepSteg through extensive experiments and comparisons with traditional methods. In our proposed method, the CNN network analyzes the robustness of the watermarked image and updates the weight. Many challenges are solved without human visualization, a deep learning method able to extract features by itself [19]. The challenges posed by low-light conditions in capturing clear and vibrant images also get solved by leveraging advanced techniques within deep learning [20]. After 2006 the deep learning field has been used in watermarking in various ways and it is mostly used for security purposes [21], [22], [23], [24].

3 Proposed Deep Learning-Based Data Hiding Framework

3.1 Architecture Overview

The proposed framework for deep learning-based data hiding combines the power of convolutional neural networks (CNNs) with watermarking techniques. The framework consists of two main components: the watermark sequence generation and embedding algorithm, and the image watermarking based on CNN.

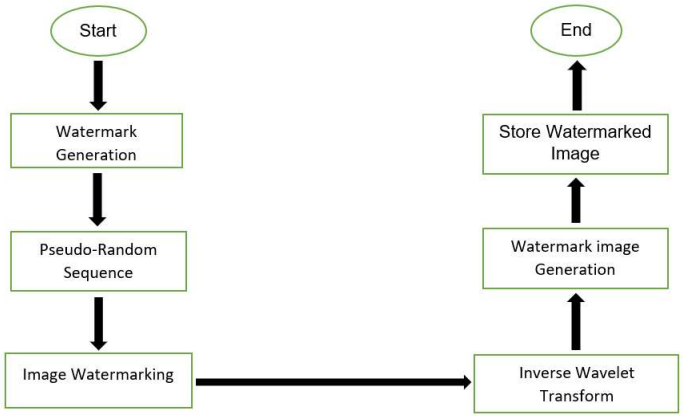

Figure 1: Watermark embedding process

3.2 Algorithm and steps of working

The watermark sequence generation and embedding algorithm aims to generate a robust and imperceptible watermark sequence that can be embedded into the host image.
The algorithm and steps for sequence generation and Watermark embedding are given below.

3.2.1 Watermark Generation:

- **Load and Convert Watermark Image to Grayscale**:
  Loading the watermark image is the first step. Converting it to grayscale ensures that the watermark is in a single-channel format, which is typically used for watermarking. Grayscale images simplify further processing and

ensure that the watermark is not influenced by colour variations.
$A = rgb2gray(imresize(imread(`watermark.png'), [512, 512]))$.

- **Perform Singular Value Decomposition (SVD) on the Watermark**:
  SVD is used to decompose the watermark image into its singular values (Sw) and singular vectors (Uw and Vw). SVD is a powerful mathematical tool for breaking down an image into its fundamental components. It's important because it allows you to manipulate the watermark in a manner that is robust and imperceptible when embedded.
  $A = Uw \cdot Sw \cdot Vw$.

- **Calculate Median Values for Uw and Vw**:
  Calculating the median values of Uw and Vw helps determine appropriate binarization thresholds. These thresholds are used to convert the singular vectors into binary form, which is essential for the watermark embedding process. The median values are often chosen because they are robust to outliers and ensure a balanced binarization.

- **Binarize Uw and Vw using Calculated Thresholds**:
  Binarization involves converting the continuous values of Uw and Vw into binary values (0s and 1s) based on the calculated thresholds. This step simplifies the data representation and prepares the singular vectors for further processing.
  $IUw(i, j) = Uw(i, j) \leq u, IVw(i, j) = Vw(i, j) \leq v$

- **Perform XOR Operation to Obtain Sequence G**:
  The XOR operation between IUw and IVw combines the binary representations of Uw and Vw to create a new binary sequence, G. This sequence carries information about the watermark and will be further processed and embedded into the host image.
  $G(i, j) = IUw(i, j) \oplus IVw(i, j)$

- **Generate Pseudo-Random Sequence K**:
  The generation of a pseudo-random sequence K is important for adding a layer of security and randomness to the watermark embedding process. This sequence is of the same length as G and ensures that the watermark embedding is not easily predictable.

- **Perform XOR Operation Between G and K to Obtain Watermark Sequence W**:

The final watermark sequence W is obtained by performing an XOR operation between sequence G and pseudo-random sequence K. This step further enhances the security of the watermark and makes it more robust against attacks.
$W = G \oplus K$

3.2.2 Image Watermarking:

For each image in the batch:–

- **Load Host Image and Convert to Grayscale**:
  Converting the host image to grayscale simplifies the watermark embedding process by reducing the image to a single channel, making it easier to manage.
  $I = rgb2gray(imresize(images(:,:,:,i),[512,512]))$.

- **Apply Discrete Wavelet Transform (DWT)**:
  DWT is used to decompose the host image into its constituent components: LL (low frequency), LH (horizontal detail), HL (vertical detail), and HH (diagonal detail). This decomposition is fundamental for watermark embedding because it allows you to focus on a specific frequency bands for embedding, which can make the watermark more robust and less perceptible.
  $dwt2(I,`haar') = [LL, LH, HL, HH]$.

- **Perform SVD on HH Component**:
  The SVD is applied to the HH component of the host image. This is typically the component that is modified to embed the watermark. SVD helps to decompose HH into its singular vectors (Uh and Vh) and singular values (Sh). By modifying Sh with information from the watermark, you can embed the watermark while preserving the structure of the HH component.

- **Modify Sh Using Watermark Singular Values**:
  The Sh values are modified using information from the watermark's singular values (Sw). This modification process allows the watermark to be embedded into the host image's high-frequency details while minimizing perceptual changes to the image.

- **Reconstruct Modified HH Component (IHH)**:
  After modifying the singular values of the HH component, you can use SVD to reconstruct the modified HH component (IHH). This step ensures that the watermark is integrated into the host image while maintaining the image's structural integrity.

- **Perform Multi-Level DWT on LL Component**:
  The LL component of the host image is further decomposed using multi-level DWT (in this case, $LL2$, $LL3$, and $LL4$ are obtained). This multi-level decomposition provides more frequency bands for embedding the watermark, allowing for greater robustness and imperceptibility.

3.2.3 Watermark Embedding, IDWT and Watermarked Image generation:

- **Reshape $LL4$ and $HH4$ into One-Dimensional Arrays**:
  $LL4$ and $HH4$ are components of the host image obtained after multi-level DWT. Reshaping them into one-dimensional arrays simplifies further processing and embedding operations.

- **Combine $LL4$ and $HH4$ into $LLHH$**:
  This step combines the low-frequency ($LL4$) and diagonal detail ($HH4$) components into a single array, $LLHH$. The watermark is typically embedded in both low-frequency and high-frequency components to ensure robustness and imperceptibility.

- **Convert $LLHH$ Values to Positive and Track Signs in SLH**:
  Converting $LLHH$ values to positive ensures that the watermark can be embedded without affecting the sign of the host image components. The sign of $LLHH$ values is tracked in SLH for later use in the inverse embedding process.
  Conversion to positive: $LLHH = abs(LLHH)$

- **Resize and Reshape the Watermark Sequence W**:
  Resizing and reshaping the watermark sequence W to match the size of $LLHH$ ensures that the watermark is properly aligned for embedding. This step helps maintain the consistency of the watermark within the host image.

- **Extract the 10th Bit from Binary Representation of $LLHH$ Values**:
  This step retrieves the 10th bit from the binary representation of $LLHH$ values. This bit will be replaced by the corresponding bit from the watermark sequence W.
  Extracting 10th bit: $bits = bin\_str(:, 10)$
- **Replace Extracted Bits with Corresponding Bits from W**:
  The 10th bit extracted from $LLHH$ is replaced with the corresponding bit from the watermark sequence W. This bit replacement is a key part of the watermark embedding process.
  Replacement of bits: $bits\_int(1 : length(W)) = W$
- **Convert Modified Binary Values back to $Integer$ Values**:
  After replacing the 10th bit, the modified binary values need to be converted back to $Integer$ values. This step ensures that the embedded watermark is in a compatible format for further processing.

  Integer conversion: $Integer = bin2dec(bin\_str)$
- **Combine $Integer$ and $LLHH$ using SLH to get $ALLHH$**:
  Finally, the $Integer$ values are combined with $LLHH$ using the sign information stored in SLH. This step ensures that the watermark is embedded correctly with the correct sign information.

3.3 Image Watermark Based on CNN

To enhance the robustness and security of the watermarking process, the proposed framework incorporates a convolutional neural network (CNN) as part of the image watermarking component.

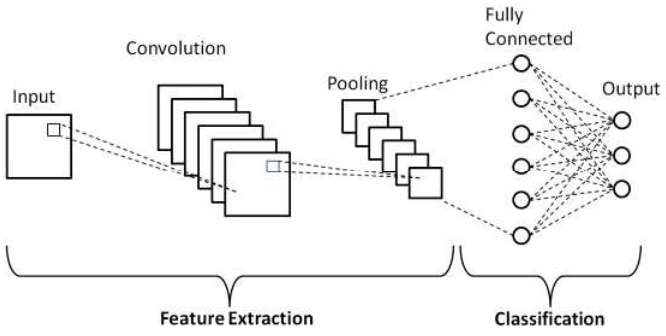

Figure 2: CNN structure

The watermarking process is divided into blocks, typically with a size of 16×16 pixels. The host image and the watermark are divided into corresponding blocks, and a CNN is constructed.

Using different keys, multiple watermark images are generated as training data for the CNN. The divided host image blocks are used as input to the CNN, which adjusts its weights based on the training data. The output of the CNN is used to modify the pixel values of the watermarking images.

By incorporating CNNs into the watermarking process, the framework leverages the ability of deep learning models to learn complex features and patterns, thereby improving the robustness and security of the data-hiding process. The results are analyzed by using a Kaggle dataset [31].

3.4 Network Training and Optimization

The CNN in the proposed framework undergoes training and optimization to improve its performance in watermarking.

During the training process, host images and corresponding watermark images are used to train the CNN. The network adjusts its weights through iterative optimization algorithms, such as backpropagation and gradient descent, to minimize the difference between the predicted watermark and the original watermark.

The trained CNN is then applied to embed the watermark into new host images, ensuring robustness against various attacks and maintaining imperceptibility.

4 Results and Discussion

We utilize a dataset comprising four original 512 × 512 grayscale images for our experimental evaluation. The images are named ”Plane”, ”Lena”, ”Man”, and ”House” as depicted in Figure 3 below. Additionally, we employ a logo watermark, represented by a 512 × 512 binary image, as illustrated in Figure 4.

These selected images offer a diverse range of content and characteristics, allowing us to assess the performance and robustness of our deep learning-based data-hiding framework across different scenarios. The grayscale nature of the images simplifies the embedding process, enabling a focused evaluation of the framework’s effectiveness in preserving watermark integrity and visual quality.

Figure 3: Original Images

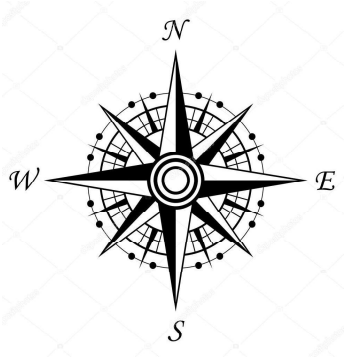

Figure 4: Logo Watermark

4.1 Performance Metrics

To evaluate the performance of the deep learning-based data-hiding framework, several performance metrics are employed. These metrics provide quantitative measures of the framework's effectiveness, robustness, and quality of the embedded watermarks. The following performance metrics are utilized:

Peak Signal-to-Noise Ratio (PSNR): PSNR measures the quality of the watermarked image by comparing it with the original (non-watermarked) image. Higher PSNR values indicate better quality and preservation of image details. The PSNR formula is given by:

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) \tag{51.1}$$

Where:

- MAX_I is the maximum possible pixel value of the image.
- MSE is the Mean Squared Error between the original and reconstructed

image.

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2 \tag{51.2}$$

Structural Similarity Index (SSIM): SSIM evaluates the similarity between the watermarked image and the original image based on perceived image quality, luminance, contrast, and structural information. Higher SSIM values indicate better similarity and preservation of image content.

4.2 Simulation Results

Table 1: Comparison Capacity and PSNR, SSIM

Image name	Method	Capacity	PSNR(dB)	SSIM
lena	Ours	512x512	43.60	0.9858
	Ref.[15]	512x512	21.56	
	Ref.[16]	512x512	18.52	
man	Ours	512x512	43.13	0.9876
	Ref.[15]	512x512	2.73	
	Ref.[16]	512x512	20.43	
plane	Ours	512x512	44.37	0.9861
	Ref.[15]	512x512	24.19	
	Ref.[16]	512x512	18.03	
house	Ours	512x512	48.48	0.9929
	Ref.[15]	512x512	20.6	
	Ref.[16]	512x512	17.38	

The results demonstrate the effectiveness of the deep learning-based framework in achieving high-quality watermark embedding while maintaining perceptual invisibility and robustness against various attacks. The analysis discusses the trade-offs between imperceptibility and robustness, identifies potential areas for improvement, and highlights the advantages of using deep learning models over traditional data-hiding techniques. Simulation result for watermarked image and its corresponding SSIM.

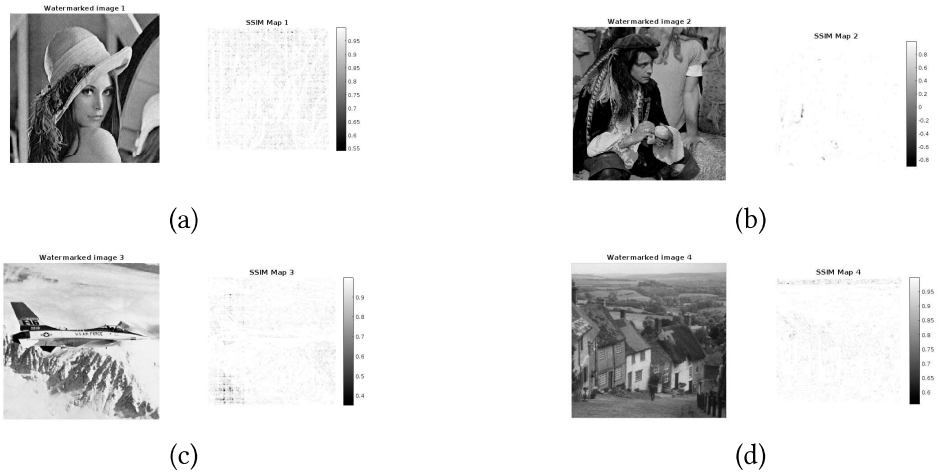

Figure 5: (a),(b),(c) and (d) are simulation results for watermarked image and its corresponding SSIM

5 Conclusion & Scope of Future Work

In conclusion, this paper has presented a deep learning-based data-hiding framework that utilizes convolutional neural networks for embedding secret information within multimedia content. The experimental evaluation demonstrated the effectiveness and robustness of the proposed framework, showcasing its ability to preserve watermark integrity and visual quality across different scenarios.

The real-world applications of deep learning-based data hiding span secure communication systems, digital forensics, and steganalysis. These applications highlight the importance and relevance of developing advanced techniques to secure and protect multimedia content in various domains.

While there are limitations and areas for future enhancements, deep learning-based data hiding holds great potential for ensuring secure communication, protecting intellectual property, and combating unauthorized data dissemination. Continued research and advancements in this field will contribute to the development of more sophisticated and reliable information-hiding techniques in the future. While deep learning-based data hiding shows promise, there are still some limitations to consider. One limitation is the potential vulnerability to advanced attacks or adversarial attempts to extract or manipulate the hidden information. Future research should focus on enhancing the robustness and security of the embedding and extraction processes to withstand such attacks.

Additionally, the computational complexity and resource requirements of deep learning models can be a challenge, particularly in real-time or resource-constrained scenarios. Future enhancements could explore techniques for optimizing the efficiency and speed of the data-hiding framework without compromising its performance.

Furthermore, the exploration of multi-modal data hiding, such as combining image and audio or text, can open up new avenues for secure communication and information hiding. Integrating multiple modalities using deep learning techniques can provide more robust and resilient data-hiding solutions.

References

[1] Tong, X., Liu, Y., Zhang, M. and Chen, Y. (2013). A novel chaos-based fragile watermarking for image tampering detection and self-recovery. *Signal Processing: Image Communication*, 28(3):301–308.

[2] Mishra, A., Agarwal, C., Sharma, A. and Bedi, P. (2014). Optimized grayscale image watermarking using dwt-svd and firefly algorithm. *Expert Systems with Applications*, 41(17):7858–7867.

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

[4] Vafaei, M., Mahdavi-Nasab, H. and Pourghassem, H. (2013). A new robust blind watermarking method based on neural networks in wavelet transform domain. *World Applied Sciences Journal*, 22(11):1572–1580.

[5] Makbol, N.M. and Khoo, B.E. (2014). A new robust and secure digital image watermarking scheme based on the integer wavelet transform and singular value decomposition. *Digital Signal Processing*, 33:134–147.

[6] Lu, Z.M. and Guo, S.Z. (2017). Chapter 3 - Lossless Information Hiding in Images on Transform Domains. *Lossless Information Hiding in Images, Syngress*, 143–204.

[7] Bhatnagar, G. and Raman, B. (2009). A new robust reference watermarking scheme based on DWT-SVD. *Computer Standards & Interfaces*, 31(5):1002–1013.

[8] Mehto, A. and Mehra, N. (2016). Adaptive Lossless Medical Image Watermarking Algorithm Based on DCT& DWT. *Procedia Computer Science*, 78:88–94.

[9] Yedroudj, M. (2020). Steganalysis and steganography by deep learning. *Université Montpellier*.

2023. In Satyasai Jagannath Nanda & Rajendra Prasad Yadav (eds.), *Data Science and Intelligent Computing Techniques*, 551–567. Computing & Intelligent Systems, SCRS, India. https://doi.org/10.56155/978-81-955020-2-8-51

[10] Singh, R. and Ashok, A. (2021). An optimized robust watermarking technique using CKGSA in frequency domain. *Journal of Information Security and Applications*, 58.

[11] Alzahrani, A. (2022). Enhanced Invisibility and Robustness of Digital Image Watermarking Based on DWT-SVD. *Applied Bionics and Biomechanics*, 1–13.

[12] Moosazadeh, M. and Ekbatanifard, G. (2019). A new DCT-based robust image watermarking method using teaching-learning-Based optimization. *Journal of Information Security and Applications*, 77:28–38.

[13] Kumari, R.R., Vijaya Kumar, V. and Naidu, K. (2023). Digital image watermarking using DWT-SVD with enhanced tunicate swarm optimization algorithm. *Multimedia Tools and Applications*, 82:1–21.

[14] Tavakoli, A., Honjani, Z. and Sajedi, H. (2023). Convolutional neural network-based image watermarking using discrete wavelet transform. *Int. j. inf. tecnol.*, 15:2021–2029.

[15] Nguyen, T., Duong, D.M. and Duc, D.A. (2015). Robust and high capacity watermarking for image based on DWT-SVD.

[16] Makhloghi, M., Akhlaghian Tab, F. and Danyali, H. (2011). Robust blind dwt based digital image watermarking using singular value decomposition. *International Journal of Innovative Computing, Information and Control*, 8(7):219–224.

[17] Vaidya, S.P. (2023). Fingerprint-based robust medical image watermarking in hybrid transform. *Vis Comput*, 39:2245–2260.

[18] Khandelwal, J. and Sharma, V. (2023). Extensive dual tree complex wavelet transform-based image steganography using SVD and CNN subspace. *Journal of Discrete Mathematical Sciences & Cryptography*, 26:617–627.

[19] Zhong, Y., Zhang, S., He, R., Zhang, J., Zhou, Z., Cheng, X., Huang, G. and Zhang, J. (2019). A Convolutional Neural Network Based Auto Features Extraction Method for Tea Classification with Electronic Tongue. *Applied Sciences*, 9:2518.

[20] Singh, K.K. and Sharma, G. (2022). Deep Image Contrast Enhancement Technique for Low-Light Images. *Research Journal of Engineering Technology and Medical Sciences*, 5(4).

[21] Dhaya, R. (2021). Light weight CNN based robust image watermarking scheme for security. *Journal of information technology and digital world*, 3(2):118–132.

[22] Li, D., Deng, L., Bhooshan Gupta, B., Wang, H. and Choi, C. (2019). A novel CNN based security guaranteed image watermarking generation scenario for smart city applications. *Information Sciences*, 479:432–447.

[23] Wang, Z., Byrnes, O., Wang, H., Sun, R., Ma, C., Chen, H., Wu, Q. and Xue, M. (2023). Data Hiding with Deep Learning: A Survey Unifying Digital Watermarking and Steganography.

[24] Abbate, G., Amerini, I. and Caldelli, R. (2023). Image Watermarking Backdoor Attacks in CNN-Based Classification Tasks. *Springer Nature Switzerland*, 3–16.

[25] Chang, C.C., Hu, Y.S. and Lu, T.C. (2006). A watermarking-based image ownership and tampering authentication scheme. *Pattern Recognition Letters*, 27(5):439–446.

[26] Lin, W.H., Wang, Y.R., Horng, S.J., Kao, T.W. and Pan, Y. (2009). A blind watermarking method using maximum wavelet coefficient quantization. *Expert Systems with Applications*, 36(9):11509–11516.

[27] Sencar, H.T., Ramkumar, M. and Akansu, A.N. (2004). Data Hiding Fundamentals and Applications: CHAPTER 8 - Data Hiding Applications. *Academic Press*, 179–219.

[28] Hinton, G.E. (2007). To recognize shapes, first learn to generate images. *Progress in brain research*, 165:535–547.

[29] Chang, C.C., Tsai, P. and Lin, C.C. (2005). SVD-based digital image watermarking scheme. *Pattern Recognition Letters*, 26(10):1577–1586.

[30] Poonam and Arora, S.M. (2018). A DWT-SVD based Robust Digital Watermarking for Digital Images. *Procedia Computer Science*, 132:1441–1448.

[31] Kaggle data set: https://www.kaggle.com/datasets/pavansanagapati/images-dataset.