Coverless Image Steganography Based on Machine Learning Techniques

Teba Hassan AlHamdani[1,*], Suhad A. Ali[1], Majid Jabbar Jawad[1]

[1]Department of Computer Science, College of Science for Women, University of Babylon, Babylon, Iraq

Emails: teba.alhamadani.gsci145@student.uobabylon.edu.iq; wsci.suhad.ahmed@uobabylon.edu.iq; wsci.majid.jabbar@uobabylon.edu.iq

Abstract

Image steganography is a technique used to conceal secret information within digital images in such a way that the existence of the hidden data is not perceptible to the human eye. This method leverages the vast amount of data contained in image files, embedding the secret message by altering certain pixel values in a manner that is undetectable. The primary goal of image steganography is to ensure that the embedded information is secure and invisible, maintaining the original image's appearance and quality. Applications of image steganography include secure communication, digital watermarking, and copyright protection. Advanced methods often employ complex algorithms and machine learning models to enhance the robustness and imperceptibility of the hidden data, making it resistant to detection and manipulation.. The main idea of the proposed work is to utilize features extracted from images to construct a Hash Table, which will be employed for concealing and revealing a secret message. Since the same CNN model and input image (i.e., cover image) produce identical features, even if the cover image is slightly affected by noise, the same features (and consequently the same Hash Table) will be generated. The work demonstrated promising results in regenerating images when the cover image is slightly affected. However, as the noise level increases on the cover image, the regenerated images begin to lose more details.

Keywords: Image steganography; Coverless; Deep learning; Encryption; Watermarking

1. Introduction

Since the advent of the Internet, one of the most critical aspects of information technology and communication has been the security of information. Steganography, the art of hiding secret data in plain sight, plays a significant role in this domain. Although it might seem counterintuitive, it is remarkably effective. Steganography conceals information within seemingly innocent files, such as images, music, or other random data. Unlike cryptography, which scrambles data, steganography hides the very existence of the message, making it a form of covert communication. The importance of data steganography can be summarized as follows [1]:

- Covert Communication**: Steganography enables hidden communication within seemingly innocuous files. Unlike cryptography, which scrambles data, steganography conceals the very existence of the message. This covert communication is essential for various purposes, including espionage, intelligence, and privacy.
- Security and Deception**: By embedding messages within images, audio files, or other digital content, steganography ensures secrecy and deception. It focuses on data hiding rather than encryption. Attackers and penetration testers often use steganography creatively to share information or carry out malicious actions.
- Advantages over Cryptography**: While cryptography ensures privacy by obscuring the content of a message, steganography goes further by hiding the message itself. This makes detection more difficult, as the presence of a hidden message is not immediately apparent.

On the other hand, Convolutional Neural Network (CNN), also known as ConvNet, is a specialized type of deep learning algorithm primarily designed for tasks that involve object recognition. These tasks include image classification, detection, and segmentation. Importance of CNNs could be summarized into:

DOI: https://doi.org/10.54216/JCIM.150214
Received: May 15, 2024 Revised: July 19, 2024 Accepted: October 30, 2024

- Feature Extraction: CNNs autonomously extract features at a large scale, bypassing the need for manual feature engineering. This enhances efficiency by allowing the network to learn relevant patterns directly from the data.
- Translation-Invariant Characteristics: The convolutional layers grant CNNs the ability to identify and extract patterns from data regardless of variations in position, orientation, scale, or translation.
- Pre-Trained Architectures: Models like VGG-16, ResNet50, Inceptionv3, and EfficientNet have demonstrated top-tier performance. These pre-trained architectures can be fine-tuned for new tasks with relatively little data.
- Versatility: Beyond image classification, CNNs can be applied to other domains such as natural language processing, time series analysis, and speech recognition.

In this study, a CNN will be employed to encrypt data (i.e., the secret message) before embedding it in the stego image.

The remainder of paper is handled in the following manner: Section 2 illustrates the general form of image steganography. Section 3 explains some previous works related to coverless image steganography. Section 4 presents the conclusions.

2. Related works

Numerous studies have summarized coverless image steganography, categorizing methods into two main groups: deep learning features CIS methods and traditional features CIS methods.

A. Deep learning features CIS methods

In their 2019 work, Liu et al. [2] proposed a method to retrieve a limited number of images using deep learning. They used the DenseNet convolutional neural network to extract features from image datasets with supervised learning. The retrieved images serve as information carriers. These images are divided into 4 × 4 sub-blocks for block discrete wavelet transform (DWT). The DWT coefficients are calculated from the low-frequency components, and the coefficients are organized into robust feature sequences through Zigzag scanning between blocks. The secret information is segmented to match the length of the feature sequence, and an inverted index is created, including the feature sequence, block positions, DWT coefficients, and image paths. Images with feature sequences matching the secret information segments are selected as carriers through the index.

To improve the resilience and precision of the method in reference [2], Tan et al. proposed a coverless information hiding method using deep learning [3]. Coverless information hiding has become a major focus in information security. To achieve fast and accurate real-time image matching, the authors generate a robust hash sequence using the feature sequence, DC, and location via Discrete Cosine Transform (DCT). They establish an inverted index structure with this hash sequence for efficient image matching. During transmission, stego-images are matched and transmitted through feature retrieval. Upon reception, the secret image is reconstructed by extracting similar blocks from the received stego-images and assembling them based on location information.

In 2022, X. Liu and colleagues [4] introduced an effective coverless steganography method using minimal mapping images. They extracted ring statistics features from images to ensure the distinctiveness and resilience of the mapping features. A chaotic system was used to scramble image features, facilitating the mapping of confidential information. This method reduces the need for numerous images and prevents inaccurate mapping by using a small set of mapping images. These images are categorized using ResNet101-based classification to correspond to multiple instances of secret information with diverse scrambled attributes.

The mentioned works used Deep Learning techniques to select the number of images used to represent the hash table. Another set of works (will be explained below) used Deep Learning in generating the hash sequence itself.

In 2020, Liu et al. [5] introduced a coverless image steganography (CIS) algorithm using DenseNet feature mapping. This method incorporates deep learning to extract high-dimensional CNN features, which are then converted into hash sequences. To expedite the search for hidden information, a binary tree hash index is constructed from the sender's side. Matched images are then transmitted. At the receiver's end, the secret information is retrieved by computing the DenseNet hash sequence of the cover image. Notably, the cover images remain unmodified throughout the entire steganography process.

Meng et al. [6] introduced a coverless image steganography algorithm based on a hash generation framework. This algorithm distinguishes itself by generating hash sequences using an end-to-end CNN model, which takes original images as input and outputs corresponding hash sequences, eliminating the need for auxiliary information during concealment. The model incorporates an attention mechanism and adversarial training to enhance robustness, necessitating a redesigned loss function. An index structure is also introduced to improve mapping efficiency.

DOI: https://doi.org/10.54216/JCIM.150214
Received: May 15, 2024 Revised: July 19, 2024 Accepted: October 30, 2024

Karim et al. [7] proposed a coverless image steganography approach that operates in the frequency domain. The embedding process involves several steps: first, the secret data is segmented without overlap. A collection of images is then gathered to identify suitable candidates for stego images. A robust hashing algorithm is applied to generate a hash sequence for each image, and an inverted index structure is constructed for these hash sequences. An image whose hash matches the secret data segment is then selected. Experimental tests show that this method withstands various image processing attacks, including JPEG compression, noise, low-pass filtering, scaling, rotation, median and mean filters, brightness adjustments, and sharpening.

Lou et al. [8] utilized deep learning to dynamically select appropriate carriers for real-time image data concealment. By using CNN-extracted high-level semantic features, the method achieves superior accuracy compared to low-level features. This approach eliminates the need for a designated image for embedding confidential data, instead transmitting real-time stego-images with visually similar blocks to the secret image. Online real-time images are segmented based on specific criteria, and DenseNet is used to extract high-level semantic features from each similar block. A robust hash sequence, including the feature sequence, DC, and location, is generated using Discrete Cosine Transform (DCT). An inverted index structure is created from the hash sequence for efficient real-time image matching. During transmission, stego-images are matched and sent using feature retrieval. At the receiver's end, the secret image is reconstructed by extracting and assembling similar blocks from the received stego-images based on location information.

In the work of Duan et al. [9], the secret image is processed through a generative model database to create a meaningful yet independent image. This generated image is sent to the receiver, who then uses the generative model database to produce an image visually identical to the secret image. Thus, only the meaningful image, unrelated to the secret image, needs to be transmitted. This novel coverless image steganographic scheme, based on generative models, differs from traditional methods by ensuring that the transmitted image does not contain any embedded information from the secret image, effectively avoiding detection by traditional tools.

B. Traditional features CIS methods

The initial coverless image steganography method, introduced by Zhou Z. and their team [10], involves generating an 8-bit hash sequence for each image and requires a database of at least 256 distinct images. The database is indexed by these 8-bit binary hash sequences to streamline the search. Secret data is converted into a bit string, segmented, and concealed using cover images whose hash sequences match the segments. Each cover image can conceal only 8 bits, necessitating the transmission of many images and a large database to cover all possible hash sequences.

Z. Zhou and their team [11] developed a coverless steganography method using the Bag-of-Words (BOW) technique to enhance robustness. This method conceals text information by embedding it into images using visual words. The BOW model maps text keywords to visual words derived from an image set. Each image is divided into smaller sections, and histograms of visual words are created for each sub-image. The sub-images with the highest frequencies of visual words are selected to represent the text data. Multiple sub-images are then combined to form stego images, which are used for covert communication.

Cao, Z. Zhou, and his team [12] proposed a coverless data concealing method to enhance the capacity for information hiding using "molecular structure images of material." The approach involves dividing the image into multiple blocks (sub-images) and computing the average pixel values for each block. These averages represent segments of the secret data based on the mapping between pixel value ranges and secret data segments. To further secure the technique, a pseudo-random label sequence is applied to identify the locations of the sub-images.

In 2019, Qiu et al. [13] used three feature types—Local Binary Pattern (LBP), mean pixel value, and pixel variance—to transmit secret information. They first generate a hash sequence of the original cover image using these features. A one-to-one comparison is then made between the secret information sequence and the hash sequence of the cover image. If there are mismatches, image blocks from the cover image are substituted based on the secret information, resulting in the creation of the stego image.

Yang, L. and her team [14] introduced a coverless information hiding technique to enhance capacity, based on the Most Significant Bit (MSB) of the cover image (CIHMSB). The cover image is first segmented into multiple fragments, and the average intensity of each fragment is calculated. The secret information is then represented using the MSB of the cover image. A one-to-one mapping is established between the secret information and the MSB of the image fragments based on a predetermined order (Km) agreed upon by both sender and receiver. To address channel noise during transmission, the sender includes a Mapping Flag (Kf) with the stego image. Due to potential distortions in regular channels, the technology requires more optimal image features and improved robustness for accurate decoding of the secret data.

DOI: https://doi.org/10.54216/JCIM.150214
Received: May 15, 2024 Revised: July 19, 2024 Accepted: October 30, 2024

In 2022, J. Pan and his team [15] introduced a coverless image steganography method using a two-level approach and unique arrangements to enhance feature diversity, capacity, and image quality. They developed a new encryption model for secret message security based on logical mapping, employing a look-up table and a two-level mechanism to transmit the secret message with a single carrier image. The location table is embedded into the original image using reversible information hiding to ensure both storage and security.

In 2023, R. Campbell and his team [16] proposed a coverless information-hiding method called CIHLHF, which utilizes the lowest and highest values of image fragments. The most significant bit of these values is merged with the secret data's binary form, and a predetermined key organizes the mapping. Their approach achieves higher concealment capacity compared to similar techniques.

3. Proposed system

Due to advancements in telecommunications and the widespread use of multimedia data, secret communication between a sender and receiver is necessary to protect sensitive information from third parties. Traditional image steganography techniques insert secret data into a cover image, creating a stego-image, but this leaves modification traces that make steganalysis easier. Therefore, the challenge is to conceal information without altering the carrier. This chapter introduces a coverless image steganography technique based on deep learning to achieve better capacity. It details the algorithms, procedures, and diagrams used in the proposed system.

This section explains the proposed coverless image steganography approach in detail. Figure 1 presents the main block diagram of the method, which is based on deep learning. The methodology consists of two procedures: the embedding procedure on the sender's side and the extracting procedure on the receiver's side.

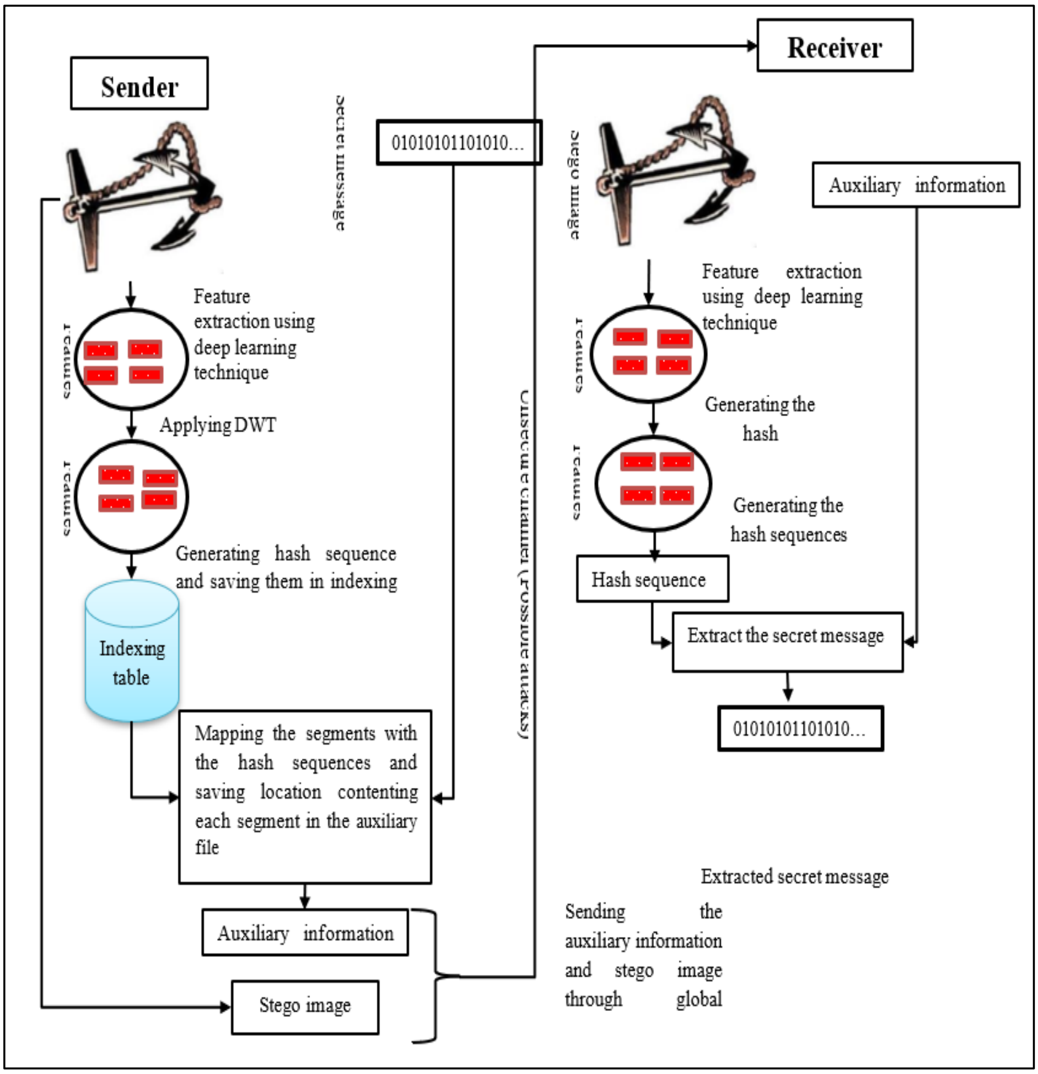

Figure 1. Block diagram of the proposed system

DOI: https://doi.org/10.54216/JCIM.150214
Received: May 15, 2024 Revised: July 19, 2024 Accepted: October 30, 2024

A. Selecting a Suitable Deep Learning Network for features extraction

Feature extraction involves using a pre-trained model as a fixed feature extractor, without modifying its weights, to extract relevant features from input data. These features can then be used in a new classifier or model tailored to the target task. In deep learning, choosing the right pre-trained model is crucial, especially for tasks like image classification, object detection, or natural language processing. The model's architecture should be adapted to match the number of classes or labels in the dataset.

In the described system, a Convolutional Neural Network (CNN) is applied to the cover image to extract features for embedding a secret message. Several pre-trained CNN networks, including Inception V3, VGG16, MobileNetV2, and ResNet50v2, are evaluated. Transfer learning, which involves using a pre-trained model as a starting point for a new classification task, is employed. This technique has proven successful, allowing for faster and more accurate training with limited labeled data by leveraging models trained on large-scale datasets like ImageNet. It reduces the need for extensive training from scratch and facilitates efficient knowledge transfer across tasks and domains.

To select the best network for feature extraction, the classification accuracy of each network on the image dataset is assessed. The network that achieves the highest classification accuracy, ResNet50v2, is chosen for feature extraction in the method.

The hyperparameters used in the training of all models are similar. The number of epochs are 20. The batch size is 15. The learning rate is 0.001. The optimization algorithm is Adam. And the loss function is Categorical Cross entropy.

B. Processes on the Sender Side

The embedding process consists of several stages. The following sections go into detail about each stage:

1. **Selecting the Appropriate Image**: The goal is to find an image that already contains all necessary information (hashes) to embed the secret data. The features of the selected image are extracted using a combination of deep learning and Discrete Wavelet Transform (DWT). Hash sequences are generated from these features using an efficient hash algorithm that can resist image processing attacks. The input image can be either gray or color.
2. **Binary Hash Sequences Generation**: A hash code, a fixed-length binary sequence, is calculated from an image based on its deep features. Each image's deep features can be expressed as a hash code, which corresponds to a specific part of the secret information. The input image is normalized to 299x299 pixels to ensure consistent feature length across different images. Several steps are involved in obtaining the hash sequence, as depicted in Figure 2.
3. **Discrete wavelet transform (DWT):** This process is used for applying the DWT on features that resultant after applying CNN. The type of generated values from DWT is float32. In the Initialization stage, the input signal x is set as the input vector. In the decomposition stage, The input signal is decomposed into approximation coefficients $a_j[n]$ and detail coefficients $d_j[n]$ for each decomposition level j. This is achieved by applying low-pass and high-pass filters to the input signal and then down sampling the resulting coefficients. And for the output stage, the transformed coefficients $a_j[n]$ and $d_j[n]$ for each decomposition level j are returned as the output of the algorithm. This algorithm performs the discrete wavelet transform on a 1D input vector, producing transformed coefficients at each decomposition level.
4. **Converting Features into Binary Form:** The resultant features after applying DWT is converted into binary form according to the equation 1.

$$f(x) = \begin{cases} 0 & if\ x < mean(X) \\ 1 & if\ x \geq mean(X) \end{cases} \quad (1)$$

Where x is the input value, and mean(X) is the mean of the input vector X.

5. **Generating a Hash Sequence from Binary Features:** Herein, all possible hash sequences are generated from resultant binary features by using overlapping. The length of each hash sequence is 8 bits. All the generated hash sequences are saved in the Hash Table (HT). The HT has three columns, the first column represents the value of the hash sequence. The second columns represents the decimal value the hash sequence in the features. The third columns represents the location of the hash sequence in the features.

DOI: https://doi.org/10.54216/JCIM.150214
Received: May 15, 2024 Revised: July 19, 2024 Accepted: October 30, 2024

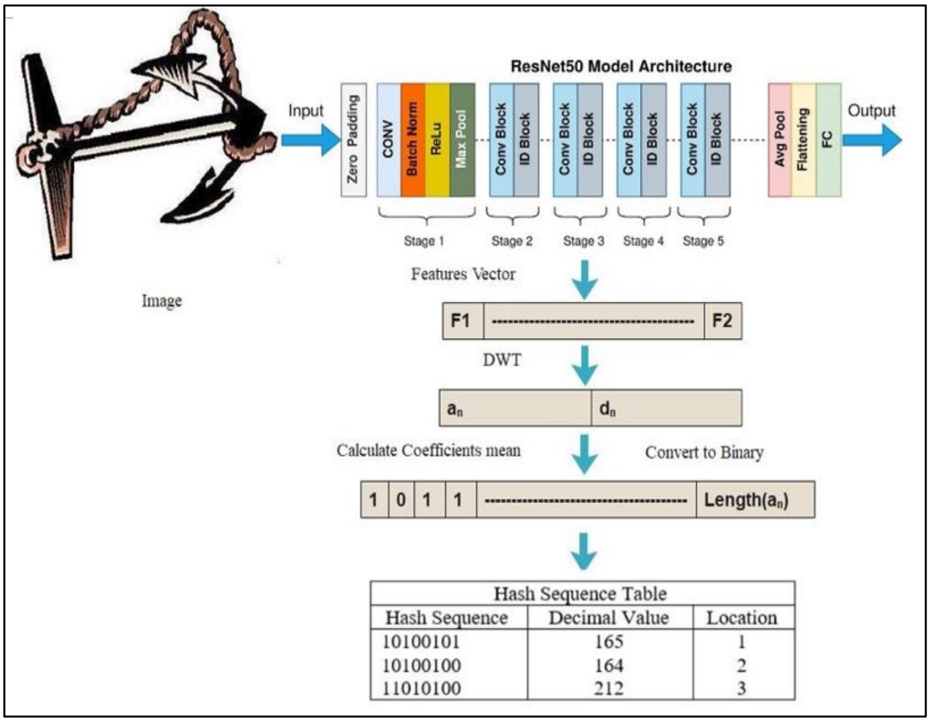

Figure 2. Generating Hash Sequenc Table from input image

6. **Building an Index Table Depending on a Hash Table:** For speeding up the time during embedding the secret message, an inverted table has been building for each sequence that have the corresponding block and location number that matched for it. The building of index table depends on the hash table. All the hash sequences are indexed in the database according to their values.
7. **Embedding Operation:** In this step, the mapping between the secret message and index table is done. The secret message is matched with the value which is found in the index table. The location of the matching for each segment are saved in the auxiliary information file. Figure 3 shows the emmbeding operation related to secret image.

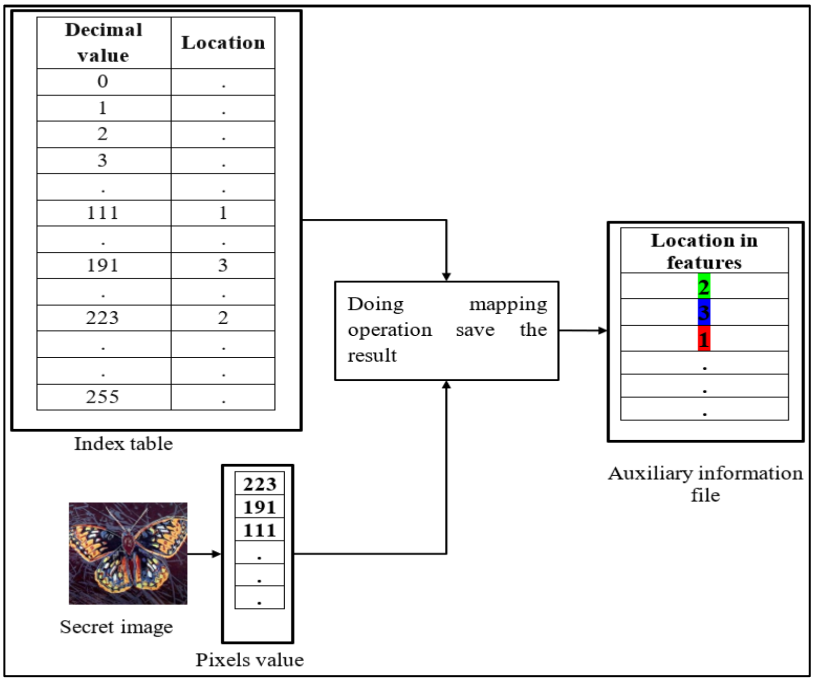

Figure 3. Image embedding operation

DOI: https://doi.org/10.54216/JCIM.150214
Received: May 15, 2024 Revised: July 19, 2024 Accepted: October 30, 2024

C. Processes on the Receiver Side

On the receiver's side, a number of processes are carried out to decode the encrypted message, which are:

1. **Applying Deep Learning Technique:** This process is used for applying the CNN network on stego image for extracting the features which will be used for extraction the secret message. Herein the CNN network (i.e. ResNte50) is used for extracting the features.
2. **Applying DWT on the Extracted Features:** This process is used for applying the DWT on features that resultant after applying CNN.
3. **Converting Features into Binary Form:** The resultant features after applying DWT is converted into binary form according to the question 1.
4. **Generating a Hash Sequence from Binary Features:** Herein, all possible hash sequences are generated from resultant binary features by using overlapping. The length of each hash sequence is 8 bits. All the generated hash sequences are saved in the Hash Table (HT). This HT is similar to the HT built in the sender side to encrypt the data.
5. **Extraction Operation:** The secret message is extracted by matching the location in the auxiliary file information with the generated hash sequences in the hash table one by one. Later, all extracted hash sequences are merged for getting the secret message. Figure 4 shows the extraction operation on the receiver side.

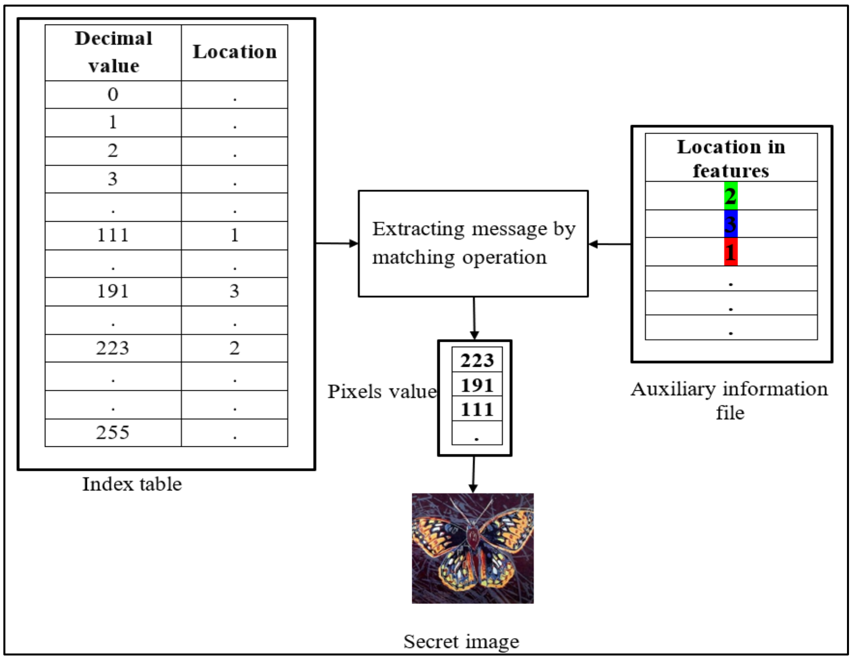

Figure 4. Image extraction operation

4. CNN Results

In this section, the results obtained from raining and testing different CNN models for the purpose of extracting features from images, are discussed. The proposed system employs various deep learning techniques, including Inception V3, VGG16, MobileNetV2, and ResNet. Figure 5 shows the training process of the proposed CNN models. And Figure 6 shows the validation process of the proposed CNN models. Notice that the dataset is divided into 80% training (7356 images belonging to 102 classes) and 20% validation (1788 images belonging to 102 classes).

The obtained validation accuracies are 82.05%, 74%, 85.29%, and 88.37% for the models Inception V3, VGG16, MobileNet2, and ResNet50 V2 respectively.

Since ResNet50 V2 get the best results, it is used as the CNN model used in the proposed system.

DOI: https://doi.org/10.54216/JCIM.150214
Received: May 15, 2024 Revised: July 19, 2024 Accepted: October 30, 2024

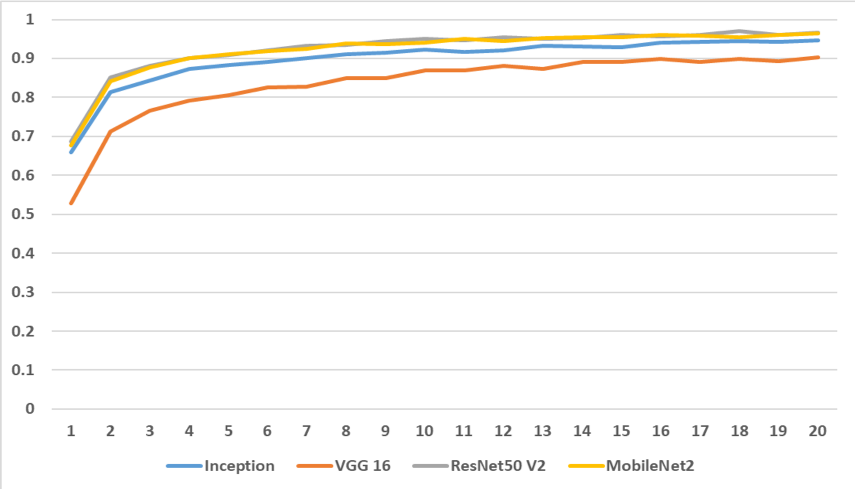

Figure 5. the training process of the proposed CNN models

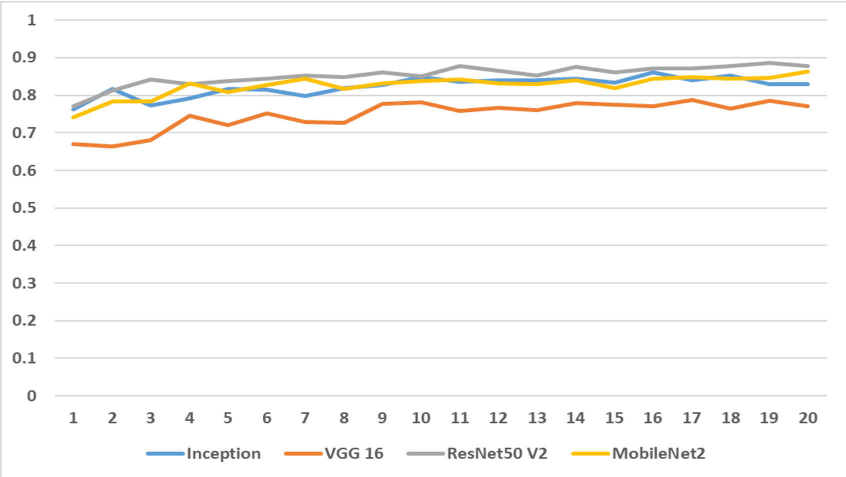

Figure 6. The validation process of the proposed CNN models

5. Results of resisting attacks

In this section, the results of the proposed method against varios attacks on the cover image will be disscussed. To test the propsoed system, the cover image will recive a difrent types of noise atttacks. This will lead to the deformation of the Hash Table, and hence the regenerated secret message (image). Therfore the more the generated image is similar to the original image, the more accurate the proposed system is.

To calculate how much similar the generated image with the original image is, three evaluation approaches will be used.

The first one is Structural Similarity Index Measure (SSIM). Which is given in the equation 2.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y+C_1)(2\sigma_{xy}+C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)} \quad (2)$$

Where: μ_x and μ_y are the average of *x* and *y* respectively. σ_x^2 and σ_y^2 are the variances of *x* and *y* respectively. σ_{xy} is the covariance of *x* and *y*. C_1 and C_2 are two variables to stabilize the division with weak denominator values.

The second measure is Bit Error Rate (BER). Which is given in equation 3.

$$BER = \frac{N_{Errors}}{N_{Total}} \quad (3)$$

Where: N_{Errors} is the number of bits received in error. N_{Total} is the total number of bits transmitted.

And the third evaluation measure is Normalized Correlation (NC). Which is given in equation 4.

$$NC(i,j) = \frac{\sum_{xy}(I(x+i,y+j)-\mu I_{i,j})(T(x,y)-\mu T)}{\sqrt{\sum_{xy}(I(x+i,y+j)-\mu I_{i,j})^2 \sum_{xy}(T(x,y)-\mu T)^2}} \quad (4)$$

DOI: https://doi.org/10.54216/JCIM.150214
Received: May 15, 2024 Revised: July 19, 2024 Accepted: October 30, 2024

Where $NC(i,j)$ is the normalized correlation coefficient at position (i,j)). $I(x+i,y+j)$ is the intensity of the larger image at position $(x+i,y+j)$. $NC(i,j)$ is the intensity of the template image at position (x,y). $\mu I_{i,j}$ is the mean intensity of the larger image region that overlaps with the template at position (i,j). μT is the mean intensity of the template image. The values of $NC(i,j)$ will range from -1 to 1, where 1 indicates a perfect match, 0 indicates no correlation, and -1 indicates a perfect inverse correlation.
Table 1 shows the type of noise, the setup used in the noise, SSIM, BER, NC, and the generated image.

Table 1: the images generated from the noises attacked on the cover image

Noise Type	Setup	SSIM	BER	NC	Image
No attack	None	1	0	1	
Average attack	Kernel size				
	3	0.974	0.107	0.989	
	5	0.964	0.11	0.983	

DOI: https://doi.org/10.54216/JCIM.150214
Received: May 15, 2024 Revised: July 19, 2024 Accepted: October 30, 2024

	7	0.961	0.111	0.982	
	9	0.957	0.112	0.979	
Gaussian attack	SD				
	0.1	0.974	0.107	0.989	

DOI: https://doi.org/10.54216/JCIM.150214
Received: May 15, 2024 Revised: July 19, 2024 Accepted: October 30, 2024

	0.3	0.977	0.106	0.99	
	0.5	0.913	0.116	0.976	
	0.7	0.784	0.131	0.928	

DOI: https://doi.org/10.54216/JCIM.150214
Received: May 15, 2024 Revised: July 19, 2024 Accepted: October 30, 2024

	1	0.678	0.14	0.875	
Gaussian low pass attack	Kernel size				
	1	0.96	0.111	0.981	
	3	0.958	0.112	0.979	

DOI: https://doi.org/10.54216/JCIM.150214
Received: May 15, 2024 Revised: July 19, 2024 Accepted: October 30, 2024

	5	0.899	0.121	0.965	
	7	0.758	0.132	0.914	
Median filter attack	Kernel size				
	1	0.974	0.107	0.989	

DOI: https://doi.org/10.54216/JCIM.150214
Received: May 15, 2024 Revised: July 19, 2024 Accepted: October 30, 2024

	3	0.971	0.107	0.987	
	5	0.96	0.112	0.98	
	7	0.948	0.114	0.975	
Salt and pepper attack	Percentage				

DOI: https://doi.org/10.54216/JCIM.150214
Received: May 15, 2024 Revised: July 19, 2024 Accepted: October 30, 2024

	0.01	0.952	0.114	0.98	
	0.03	0.793	0.132	0.933	
	0.05	0.77	0.134	0.919	

DOI: https://doi.org/10.54216/JCIM.150214
Received: May 15, 2024 Revised: July 19, 2024 Accepted: October 30, 2024

	0.1	0.664	0.141	0.868	
	0.15	0.69	0.141	0.88	
Sharpened attack	Kernel size				
	1	0.974	0.107	0.989	

DOI: https://doi.org/10.54216/JCIM.150214
Received: May 15, 2024 Revised: July 19, 2024 Accepted: October 30, 2024

	3	0.964	0.11	0.984	
	5	0.962	0.111	0.983	
	7	0.949	0.114	0.979	

6. Conclusion

The basic premises of the proposed work is to use the features extracted from images to build a Hash Table. This Hash Table will be used to hide and reveal the secret message. And since by using the same CNN model with the same input image (i.e. cover image), the same features will be generated. Therefore, even if the cover image is effected by little noise, the same features (and hence Hash Table) will be generated. The work showed a good results in regenerating the images when the cover image is effected slightly. However, when the level of noise is high on the cover image, the regenerated images start to loss more details.

DOI: https://doi.org/10.54216/JCIM.150214
Received: May 15, 2024 Revised: July 19, 2024 Accepted: October 30, 2024

References

[1] M. Hussain, A. Abdul Wahab, N. Javed, and K.-H. Jung, “Hybrid Data Hiding Scheme Using Right-Most Digit Replacement and Adaptive Least Significant Bit for Digital Images,” Symmetry (Basel), vol. 8, no. 6, p. 41, May 2016, doi: 10.3390/sym8060041.

[2] Q. Liu, X. Xiang, J. Qin, Y. Tan, J. Tan, and Y. Luo, “Coverless steganography based on image retrieval of DenseNet features and DWT sequence mapping,” Knowl Based Syst, vol. 192, p. 105375, Mar. 2020, doi: 10.1016/j.knosys.2019.105375.

[3] L. Tan, J. Liu, Y. Zhou, and R. Chen, “Coverless Steganography Based on Low Similarity Feature Selection in DCT Domain,” Radioengineering, vol. 32, no. 4, pp. 603–615, Dec. 2023, doi: 10.13164/re.2023.0603.

[4] X. Liu, Z. Li, J. Ma, W. Zhang, J. Zhang, and Y. Ding, “Robust coverless steganography using limited mapping images,” Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 7, pp. 4472–4482, Jul. 2022, doi: 10.1016/j.jksuci.2022.05.012.

[5] Q. Liu, X. Xiang, J. Qin, Y. Tan, and Y. Qiu, “Coverless image steganography based on DenseNet feature mapping,” EURASIP J Image Video Process, vol. 2020, no. 1, p. 39, Dec. 2020, doi: 10.1186/s13640-020-00521-7.

[6] L. Meng, X. Jiang, Z. Zhang, Z. Li, and T. Sun, “A Robust Coverless Image Steganography Based on an End-to-End Hash Generation Model,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 33, no. 7, pp. 3542–3558, Jul. 2023, doi: 10.1109/TCSVT.2022.3232790.

[7] N. A. Karim, S. A. Ali, and M. J. Jawad, “A coverless image steganography based on robust image wavelet hashing,” TELKOMNIKA (Telecommunication Computing Electronics and Control), vol. 20, no. 6, p. 1317, Dec. 2022, doi: 10.12928/telkomnika.v20i6.23596.

[8] Y. Luo, J. Qin, X. Xiang, Y. Tan, Q. Liu, and L. Xiang, “Coverless real-time image information hiding based on image block matching and dense convolutional network,” J Real Time Image Process, vol. 17, no. 1, pp. 125–135, Feb. 2020, doi: 10.1007/s11554-019-00917-3.

[9] X. Duan, H. Song, C. Qin, and M. K. Khan, “Coverless Steganography for Digital Images Based on a Generative Model,” Computers, Materials & Continua, vol. 55, no. 3, pp. 483–493, 2018.

[10] Z. Zhou, H. Sun, R. Harit, X. Chen, and X. Sun, “Coverless Image Steganography Without Embedding,” 2015, pp. 123–132. doi: 10.1007/978-3-319-27051-7_11.

[11] Z. Zhou, Y. Cao, X. Sun, and C. Gao, “Coverless Information Hiding Based on the Molecular Structure Images of Material,” Computers, Materials & Continua, vol. 54, no. 2, 2018.

[12] X. Zhang, F. Peng, and M. Long, “Robust Coverless Image Steganography Based on DCT and LDA Topic Classification,” IEEE Trans Multimedia, vol. 20, no. 12, pp. 3223–3238, Dec. 2018, doi: 10.1109/TMM.2018.2838334.

[13] A. Qiu, X. Chen, X. Sun, S. Wang, and G. Wei, “Coverless Image Steganography Method Based on Feature Selection,” Journal of Information Hiding and Privacy Protection, vol. 1, no. 2, pp. 49–60, 2019, doi: 10.32604/jihpp.2019.05881.

[14] L. Yang, H. Deng, and X. Dang, “A Novel Coverless Information Hiding Method Based on the Most Significant Bit of the Cover Image,” IEEE Access, vol. 8, pp. 108579–108591, 2020, doi: 10.1109/ACCESS.2020.3000993.

[15] J.-S. Pan, X.-X. Sun, H. Yang, V. Snášel, and S.-C. Chu, “Information Hiding Based on Two-Level Mechanism and Look-Up Table Approach,” Symmetry (Basel), vol. 14, no. 2, p. 315, Feb. 2022, doi: 10.3390/sym14020315.

[16] K. Anggriani, S.-F. Chiou, N.-I. Wu, and M.-S. Hwang, “A High-Capacity Coverless Information Hiding Based on the Lowest and Highest Image Fragments,” Electronics (Basel), vol. 12, no. 2, p. 395, Jan. 2023, doi: 10.3390/electronics12020395.

DOI: https://doi.org/10.54216/JCIM.150214
Received: May 15, 2024 Revised: July 19, 2024 Accepted: October 30, 2024