IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Review of Computer Engineering Studies
Vol. 11, No. 2, June, 2024, pp. 13-29
Journal homepage: http://iieta.org/journals/rces

Enhancing Steganography in 256×256 Colored Images with U-Net: A Study on PSNR and SSIM Metrics with Variable-Sized Hidden Images

Dilara Şener[1,2*], Selda Güney[2]

[1] Presidency of the Republic of Turkiye, Defence Industry Agency, Ankara 06420, Turkey
[2] Department of Electrical and Electronics Engineering, Başkent University, Ankara 06790, Turkey

Corresponding Author Email: dsener@ssb.gov.tr

https://doi.org/10.18280/rces.110202

Received: 11 December 2023
Revised: 31 December 2023
Accepted: 5 January 2024
Available online: 30 June 2024

Keywords:
image steganography, data hiding, U-Net architecture, deep learning, information security

ABSTRACT

In digital communications, the imperative for secure data transmission is increasingly addressed through steganography, wherein information is clandestinely embedded within various digital media. This study is concerned with the enhancement of steganographic techniques through a modified U-Net architecture, designed to embed 256×256 colored message images into identically sized cover images, thereby augmenting capacity for data concealment. The classical U-Net architecture has been adapted by the incorporation of batch normalization and residual blocks, aiming to refine the embedding and extraction processes's efficiency. The novel model, trained on the expansive ImageNet database, introduces the one cycle learning rate scheduler and the AdamW optimizer into the U-Net framework, achieving enhanced training efficiency, hastened convergence, and superior generalization. Validation was conducted through two distinct analyses: the first evaluating the impact of secret image size variations on the cover image within the steganographic process, and the second assessing model performance on three datasets—Linnaeus 5, ImageNet, and Labeled Faces in the Wild (LFW). Empirical assessments indicate that the proposed model outperforms existing deep learning-based steganographic methods, as evidenced by the attained metrics, particularly Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM). On the Linnaeus 5 dataset, embedding yielded a PSNR of 44.4656 dB and an SSIM of 0.9897, while extraction recorded a PSNR of 43.5393 dB and an SSIM of 0.9875. The ImageNet dataset saw an embedding PSNR of 45.3966 dB and an SSIM of 0.9906, with extraction values of 44.8206 dB PSNR and 0.9903 SSIM. Notably, the LFW dataset embedding resulted in a PSNR of 48.1407 dB and an SSIM of 0.9930, and extraction achieved a PSNR of 47.5296 dB and an SSIM of 0.9907. The qualitative and quantitative outcomes affirm the efficacy of the proposed method for the secure transmission of confidential imagery, with potential applications ranging from the safeguarding of medical records to the protection of sensitive data across various digital platforms.

1. INTRODUCTION

The development of network and computer technologies has greatly facilitated the flow and exchange of information in communication. However, this convenience has introduced security concerns regarding the processing as well as the protection of large amounts of data.

Certain security practices that have existed for a long time have been transferred to computer science for these reasons. Today, the content of these concepts is expanding much more rapidly than in the past. Data security encompasses numerous fields and disciplines, including cryptography, secret sharing, watermarking, and steganography [1].

Steganography is a data concealing technology that allows a sender and a recipient to communicate across open channels while transmitting cover media containing secret information, preventing perceptual detection by an observer [2]. In this method, secret information is typically embedded within a digital carrier like a digital image, audio file, or other digital mediums.

While cryptography and secret sharing have ancient roots, watermarking and steganography are relatively newer fields that gained traction with the widespread adoption and improvement of digital computers. Their focus on data security, which aims to protect data from unauthorized individuals, is a characteristic they share [3]. Cryptography is based on the changing the format of stored data and its reconstruction solely by the recipient. In this sense, secret sharing is similar to cryptography. According to this discipline, data is distributed to a predetermined number of individuals, and the original data is obtained only when a predetermined number of shares are combined.

Steganography, on the other hand, takes an entirely different path than other techniques. Its primary objective is not to change the format of the data, but rather to conceal it within seemingly unrelated data. The data used as a cover for the

hidden data should not attract the attention of unauthorized individuals; they should be unable to detect it even if they come across it [4].

The foundations of modern steganography were laid by Simmons with the introduction of the prisoner's issue. Within this problem, two prisoners communicate secretly by exchanging messages while being monitored by a warden. The challenge is to hide the communication so that the warden is unaware of the hidden message. This scenario highlights the concept of covert communication, where information is concealed within seemingly innocent data [5].

In steganography, it is possible to hide data within different file types, including digital images, text, audio or video. The concealed data can also be of a similar type, like digital images, text, audio or video [6]. Despite its existence in numerous fields and widespread application, image steganography is the driving force behind steganography. This is primarily because images are the easiest to share and transport. Images have the capacity to store larger amounts of hidden data, making it useful for concealing more information or larger files and can be viewed and shared on nearly all types of devices. In addition, it is difficult to discern that a confidential media is on the cover media [7].

There are primarily four essential characteristics of all steganographic systems: They must be imperceptible, secure, capable of hiding information and robust [8-10].

Imperceptibility stands as the foremost essential criterion in any method of data embedding. The essential feature and advantage of a steganographic method reside in its ability to effectively hide confidential information within a digital image to such an extent that it remains imperceptible to human visual perception and statistical examination [11].

Within the context of a steganographic system, the term "security" can be understood as implicitly encompassing the concepts of "un-noticeability" or "undetectability". Therefore, any steganography method's security is based on its ability to ensure that the confidential information remains undetectable through statistical analysis or removal even after an attacker has discovered it. The primary necessity in the steganographic procedure is the secure conveyance of confidential information. Ensuring security is of utmost importance in order to prevent unauthorized access to data during transmission over an open channel [12]. The primary objective of the security concept is not solely limited to maintaining the confidentiality of information, but also encompasses the prevention of information alteration or corruption, as well as the management of unauthorized access.

The embedded data capacity is the term used to describe the upper limit of data that may be hidden during a given steganographic process. An increased payload capacity entails the capability to accommodate a greater amount of concealed information, yet it can also heighten the probability of detection due to modifications made to the medium used for carrying said information. Hence, it is crucial to attain a state of equilibrium among payload capacity, privacy, and detectability within the context of steganography applications. The main goal of an effective steganographic system is to transmit the highest possible amount of information while utilizing the least amount of cover media. The embedding rate can be described as the proportion of the amount of hidden information (measured in bits) to the cover media's size [13].

The concept of robustness refers to the capability of the encoding and decoding scheme to maintain its effectiveness even in the presence of alterations to the stego image resulting from external image processing methods including resizing, scaling, and rotation [9, 10].

1.1 Literature review

Various conventional approaches, including Least Significant Bit (LSB) substitution [14, 6], Pixel Value Differencing (PVD) [15, 16], Discrete Wavelet Transform (DWT) [17, 18], and Exploiting Modification Directions (EMD) [19] are commonly employed in the field of steganography. There are also studies aimed at increasing the level of security of these traditional techniques using encryption algorithms [20-23].

The concealment efficacy of conventional techniques is limited due to the potential for visual distortion resulting from excessive pixel overload during the procedure in which the confidential data is embedded.

The efficiency of typical image steganographic techniques is enhanced when the embedding process minimizes stego-image distortion and maximizes the hiding capacity. Additionally, the effectiveness of these techniques is measured by the ability to minimize retrieval errors and ensure the security of the confidential data against unauthorized access. Advanced machine learning approaches are utilized in machine learning-based steganographic methods to achieve the aforementioned improvement in efficiency [12].

In recent times, significant growth has been observed in research interest surrounding Convolutional Neural Network (CNN) based image steganography. This heightened attention can be attributed to the greater capabilities it offers in comparison to conventional approaches [24]. The utilization of CNN models in image steganography draws significant inspiration from the encoder-decoder architecture, which is employed for image hiding and extraction. The encoder takes two inputs, namely the cover and the hidden image, and utilizes them to build the stego image. Subsequently, the decoder takes the stego image as input and produces the extracted hidden image. While the essential idea stays unchanged, researchers have employed different methodologies to investigate diverse architectural frameworks [25].

Rehman et al. [26] offered a steganography approach based on CNN that enhanced the stego image's visual quality by concealing the grayscale hidden image within specified extracted features of the color cover image. The authors provide a novel encoder-decoder structure for the steganography of images, utilizing deep learning techniques. Encoder-decoder architecture introduced in this study distinguishes itself from other methods in terms of input processing. At the input end of the encoder network, there are two branches running in parallel, designated for the cover and the message image. Through convolutional layers, features are taken from both the cover and covert images and then concatenated. The stego image is created by concatenating these features. The analysis is conducted using MNIST, CIFAR10, PASCAL-VOC12, ImageNet, and LFW datasets.

A scheme for auto encoders and decoders is presented in study [27]. In this scheme, three networks are constructed. The initial network converts the Red, Green, and Blue (RGB) pixels of the hidden image into features. The secondary network is a concealing network that conceals within the cover image the features extracted via the network of preparation. The last part is the extraction mesh, which takes the cover image and extracts the hidden image from it. There are two

types of losses that are computed in the context of this study: the loss between the stego and the cover image, and the loss between the decoded hidden and the original hidden image. The model is assessed utilizing the SSIM. The objective of this research is to conceal a hidden image of dimensions N×N×RGB within a carrier image of the same size, while minimizing any disruption to the carrier. By adopting this approach, the requirement for perfect reconstruction of the previous secret information is loosened, allowing for a trade-off between the restored secret image's quality and the stego image's quality. The ImageNet database is used for testing and training.

The encoder-decoder architecture has been suggested by Wu et al. in their studies [28, 29]. The proposed approach involves the direct acquisition of end-to-end mappings directly between the embedded and the cover medias, as well as between the hidden and the decoded medias. The utilization of Exponential Linear Unit (ELU) and Batch Normalization (BN) techniques is observed.

The authors suggest a generative steganography framework employing the autoregressive model PixelCNN, as discussed in reference [30]. The pixel distribution of the cover media is acquired through the application of a pixel CNN. Following this, the confidential data is evenly integrated into the pixel distribution by the technique of reduced sampling.

In the study [31], a convolutional neural network architecture named Image Steganography Generative Adversial Network (ISGAN) is applied for image steganography. The cover media is transformed into the YCrCb image format, consisting of Luma (Y), Chroma Red (Cr), and Chroma Blue (Cb), which is a color space used in digital image and video processing. The secret image is concealed only on the Y channel. Additionally, the secret media is transformed into a grayscale image format to reduce the payload, making this method specifically designed for concealing grayscale images. To generate the stego image, the encoder-decoder network uses the grayscale secret image and the Y channel of the cover image as inputs. By using the Y channel, only the hidden grayscale image is concealed, while the Cr and Cb channels remain unaffected as they contain all color-related data. The revealing network receives the Y channel of the stego image and uses it to create the grayscale secret image. This method has been tested on ImageNet, LFW and PASCAL-VOC datasets.

The research conducted in the study [32], deep convolutional autoencoder architecture is used. This architecture includes three main stages: preparation, embedding, and extraction. In the preparation layer, both cover and secret images are processed through a preprocessing module, with their features being extracted. The encoder portion of the concealing network comprises two convolutional layers with filter counts of 64 and 128, respectively. The decoder part consists of five convolutional layers, where the number of filters decreases progressively (128, 64, 32, 16, 8). The extraction network's encoder portion also consists of five convolutional layers, with progressively more filters (8, 16, 32, 64, 128). Similarly, its decoder part also has five convolutional layers, but with a decreasing filter count (128, 64, 32, 16, 8). In this study, COCO, CelebA, and ImageNet datasets are employed for experimentation.

Liu et al. [33] detail a process that effectively utilizes the capabilities of wavelet transform-based methods and U-Net. U-Net is an effective deep learning architecture known for its detailed feature extraction and precise data processing capabilities. This process is aimed at concealing grayscale images within colored ones. The system consists of two parts: a hidden network that embeds wavelet coefficients of secret data into an image, producing a visually appealing hidden image, and an extraction network that dissects the image into four wavelet coefficients to retrieve the original secret data image through reverse wavelet transformation. In a subsequent study referenced as Liu et al. [34] introduced a refined U-Net architecture with a smaller network scale, and in this study, they achieved a 6.3 dB increase in PSNR compared to their previous work. The network in this study is trained and tested using images from the ImageNet and PASCAL-VOC datasets.

A CNN with six layers and a U-Net architecture are proposed for concealment and extraction, respectively, in paper [35]. The trained neural network is comprised of both a concealing network and an extraction network, utilizes the ImageNet dataset for dataset for training and testing. Before sending the secret image to the recipient, the sender utilizes the concealing network to embed it, without alteration, inside another full-size image. After that, the receiver reconstructs the embedded image using the extraction network. The results are evaluated using the PSNR and SSIM measurement parameters.

Himthani et al. [24] implement U-Net, U-Net++ and V-Net encoders for image steganography. U-Net++ is a model developed to expand the structural connections of U-Net, but the additional complexity and computational demand result in higher hardware requirements and longer processing times. V-Net, while structurally similar to U-Net, is a variation designed for volumetric (3D) images, offering better results on three-dimensional data. A comparative evaluation of the efficacy of the U-Net, U-Net++ and V-Net designs is conducted using the LFW and Know Your Data datasets. These architectural techniques are used to conceal the secret image within the cover image. In addition, to extract the hidden image from the cover image, a standard, one-of-a-kind decoder is devised for every architecture. Considering the outcomes of the experiment, it is determined that the U-Net design performs more effectively than the other two architectures due to its higher embedding capacity and ability to produce stego and reconstructed secret images of higher quality. In this study, the reason for U-Net's greater success in image steganography compared to other architectures may stem from its effective architecture that balances complexity with computational efficiency, and its suitability for the specific types of images and data distributions in the datasets used.

An image-to-image steganography technique is proposed in another study that uses U-Net and involves embedding secret images into the Y channel of a cover image. This approach employs a unique loss function that combines Mean Square Error (MSE) and Perceptual Path Length (PPL) to enhance the quality of both the stego and the extracted hidden images. The architecture is tested on the LFW and PASCAL-VOC [36].

In the study [37], the U-Net++ design is utilized to embed a gray secret image into a color cover image. The hiding network, serving as an encoder, combines cover image's Y channel with the secret image, forming a 2-channel tensor as its input. The extraction network is made up of six convolutional layers, without any pooling layers. The study employs the ImageNet and LFW datasets for its analysis.

Three distinct network architectures – convolutional neural network (proposed by Baluja [27]), U-Net (designed by Duan et al. [35], and Swin Transformer – are currently being utilized for image embedding and extraction challenges in paper [38].

These structures were validated using the ImageNet dataset, and their results were compared to assess their respective efficacies.

The study introduces a method for embedding one image into another using MobileNet Convolutional Neural Network and U-Net design, with MobileNet serving as the backbone in both hiding and extracting networks within the U-Net structure [39]. The method undergoes training and evaluation on image datasets including StanfordCars, STL10, and CIFAR10. The results demonstrate the method's effectiveness in embedding and extracting images, achieving average PSNR values. When visual results are examined, cover and secret images exhibit visually detectable changes at a noticeable level.

As mentioned in the study [40], U-Net architecture was employed as the hiding network to conceals a color image into another one that is the same size. The hiding network merges a cover image with a secret image into a tensor using convolutions. It uses a module with four branches for varied feature extraction. The encoder and decoder apply 3×3 convolutions and Rectified Linear Unit (ReLU) for feature processing, creating a stego image. The extracting network uses similar convolutions and a final sigmoid function in order to extract the hidden image. ImageNet, LFW and PASCAL-VOC datasets are used conducted tests.

U-Net outperforms other architectures in steganography primarily due to its unique architecture. The symmetric paths of the U-Net design allow for accurate localization and are particularly effective for image segmentation tasks. This architecture facilitates the detailed reconstruction of images, which is crucial in steganography for embedding and retrieving data without noticeable alterations to the image.

1.2 Contributions of the research

Steganography and image processing are key research areas within information technology and digital media, and advancements in these fields are yielding significant results. Steganography enables the undetectable concealment of sensitive data, leading to new paradigms in information security, playing a critical role particularly in cyber security and covert communication strategies. Image processing technology, which involves transforming and analyzing raw image data, opens up a broad spectrum of applications including digital watermarking, content authentication, and secure facial recognition systems. The development of these disciplines enhances security and efficiency through advanced security protocols and data analysis techniques, while also offering innovative approaches to privacy and data protection.

In our study, we focus on the utilization of the U-Net design in both the data concealment and data extraction phases. The important contributions of our research are outlined as follows:

(1) The proposed method differs from spatial domain methods such as LSB and PVD, which are widely used in the field of steganography, as well as transform domain methods such as DFT and DCT. In the mentioned methods, the amount of secret data within the cover image is limited. However, in our study, the entire hidden image is distributed over the bits present on the cover image. This results in enhanced data hiding capacity and improved security, overcoming the limitations of current steganography techniques.

(2) RGB cover images with a 256×256-pixel resolution are used to hide RGB secret images. These secret images progressively increase in size, starting from 32×32 pixels, then to 64×64 pixels, followed by 128×128 pixels, and finally reaching 256×256 pixels. This progressive scaling allows us to examine the effects of hidden images of different sizes on the steganographic process. As the input message image's size increases, the complexity of embedding the image discreetly escalates, challenging our method to maintain the quality and undetectability of the steganographic output.

(3) In our research, the Linnaeus 5 dataset, which had not been explored in previous studies, was uniquely utilized to test our model. This dataset, offering a diverse range of high-quality images, provides an extensive and rigorous testing environment. The use of Linnaeus 5 ensures that the evaluation of the U-Net model encompasses a wide array of realistic scenarios, distinguishing our study from others where more homogeneous datasets may be relied upon. By this method, the credibility of the findings is enhanced, illustrating the model's effectiveness and suitability for practical steganographic applications. The employment of this dataset not only presents the U-Net model with varied real-world conditions but also establishes a novel standard in the testing approaches for steganography. Our study not only offers a detailed analysis using the Linnaeus 5 dataset but also includes a comparison with the results from the widely used ImageNet and LFW datasets in the literature. This approach illustrates the effectiveness of our findings in working across various types of databases and their reliability in preserving high visual quality and structural integrity.

(4) Differing from other studies in literature, the integration of the one cycle learning rate scheduler with the AdamW optimization algorithm in this study is grounded in their complementary strengths for enhancing U-Net training. OneCycleLR's dynamic learning rate adjustment is instrumental in achieving faster and more stable convergence, crucial for complex models like U-Net in steganography. AdamW, with its advanced weight decay feature, significantly improves the model's ability to generalize, a critical factor in ensuring accurate data embedding within images. The combination of the one cycle learning rate scheduler and the AdamW optimization algorithm in this study ensures a balance between quick learning and effective regularization. This approach is expected to lead to a model that not only learns efficiently but also maintains high accuracy and reliability.

(5) After analyzing the obtained PSNR and SSIM results, promising findings have been achieved in comparison to previous deep learning studies that used cover and secret images of the same dimensions, to the best of our knowledge. The improvement in outcomes highlights the success of the suggested technique in preserving image quality throughout the hiding and revealing process.

In our study, the architecture and algorithms employed are anticipated to offer useful insights for future research in steganography, particularly in terms of accommodating larger sizes of hidden data and achieving better metric ratios. It is thought that the contributions made will influence the development of advanced steganography methods and data hiding technologies, with a special emphasis on the unique applications of the U-Net architecture and its effectiveness in complex datasets. Especially by combining the innovative use of U-Net architecture with the one cycle learning rate scheduler and the AdamW optimization algorithm, the approach is thought to address the limitations of previous methods and contribute to new directions in research, potentially leading to more refined and effective data hiding and extraction techniques.

The remaining parts are arranged as follows: Sections 2 describes characteristics and applications of the U-Net architecture. Section 3 presents the suggested approach. Section 4 outlines datasets used and experimental setup. Section 5 provides experimental outcomes and metrics used. Conclusions are presented in Section 6.

2. THE U-NET ARCHITECTURE

U-Net is an architecture of neural networks that was developed mainly for the purpose of image segmentation [41]. Image segmentation is particularly important in medical imaging, yet traditional methods often fall short in precisely defining boundaries and efficiently handling computationally demanding situations. The U-Net architecture aims to address these fundamental challenges in the field, offering more accurate and efficient image segmentation. It primarily employs the encoding and decoding techniques to integrate the underlying and higher-level information [35].

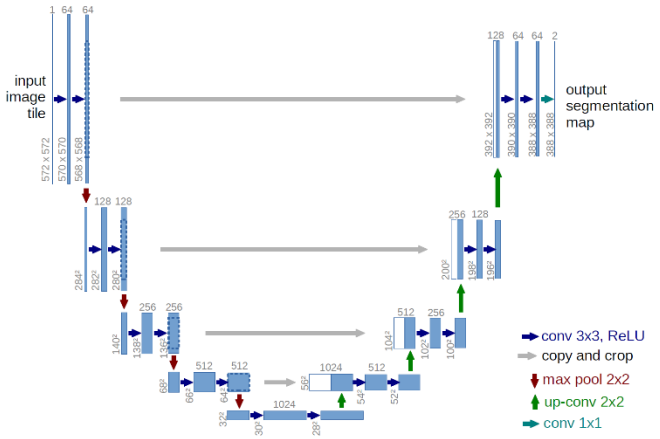


Figure 1. Basic U-Net architecture [35]

As shown in Figure 1, two pathways make up the basic framework of a U-Net architecture. The first pathway is the contracting pathway and the second pathway is an expansion pathway. The contracting pathway adheres to the conventional structure of a convolutional network. To achieve downsampling, the procedure entails applying two 3×3 convolutions iteratively, then performing a ReLU and 2×2 maximum pooling operation with 2 strides. Downsampling is used to reduce the input image's spatial dimensions progressively. This reduction helps capture high-level features and context information. During each downsampling iteration, the number of feature channels increased twofold. 'Feature channels' refer to the individual feature maps produced by different network layers. Increasing the number of feature channels allows the network to learn a richer set of features from the input data, potentially improving its ability to discriminate between different objects or regions in the image. Each stage in the expansive pathway involves several operations. The feature map is first upsampled, and then it is subjected to a 2×2 convolution, which halves the number of feature channels. Upsampling, is employed to recover the spatial resolution and produce a segmentation map that aligns with the original image dimensions. Next, a concatenation is present between the contracting pathway's corresponding cropped feature map. After that, there are two 3×3 convolutions. Each is suceeded by ReLU activation function. In the last layer of the U-Net design, the 1×1 convolution operation transforms the feature vectors, which consist of 64 components, into the format suitable for the output classes targeted by the model [42]. As implied by its name, the 1×1 convolution is an operation applied to a single-pixel-wide area in each feature map. This process compresses the high-dimensional features learned in the previous layers of the network into a more manageable format, making them usable for various visual processing tasks. The 1×1 convolution reduces the depth dimension of the network while preserving the complex feature information, thus enhancing the model's computational efficiency and its ability to produce outputs effectively with fewer parameters in the final layer.

An important characteristic of this network is the bypass connections between each level. These connections directly transfer the feature maps produced at each level of the contracting pathway to the corresponding level of the expanding pathway. Known as 'skip connections' in deep learning architectures, they enhance the network's learning efficiency. The primary function of the bypass connections is to integrate the general context information from the contracting pathway with the detailed local information from the expanding pathway. Consequently, U-Net can merge high-level features (for example, general shapes or structures) obtained from the input image with detailed local features (such as specific edges or texture information) to perform more accurate and comprehensive visual analysis. In summary, bypass connections are a crucial part of the U-Net architecture, enabling it to produce high-quality results by preserving both the general context and fine details of the input image.

In our study, the detailed description of the U-Net architecture provides a critical connection to understanding our investigation in the domain of deep steganography. The advanced image processing capabilities of U-Net are ideal for steganographic applications such as embedding and extracting hidden information within images. Particularly, U-Net's precise image analysis abilities enable the improved steganographic data hiding and retrieval processes, which are the main goals of our work. In this context, the detailed explanation of the U-Net design is essential to deeply understand the innovative aspects of our research and its contributions to the field of steganography.

3. PROPOSED METHOD

As mentioned previously, while the U-Net design has primarily been employed for medical image segmentation, the purpose of this work is to employ the U-Net architecture for steganography application. U-Net architecture offers significant advantages in steganography over traditional methods. In particular, its skip connections allow features to be directly transferred from lower to higher levels in the network, making it easier to embed and extract hidden data with greater accuracy and efficiency. U-Net's deep learning-based structure excels in learning complex visual details, thus providing higher accuracy and efficiency in steganographic processes. These features clearly demonstrate why U-Net architecture is preferred for steganography applications over traditional methods. Therefore, the U-Net architecture is employed for both the concealing and revealing of the secret image in this study.

In adapting the U-Net design for steganography, the main challenge is to embed data imperceptibly and ensure its accurate recovery. To address this, residual blocks and BN are integrated into the architecture. Residual blocks, utilized through convolutional layers, minimize information loss even in deep network structures, allowing for more precise processing of hidden data. These blocks are known to improve the learning process and data flow by directly transferring a portion of the input data to the output at each layer. This approach is especially critical in processing complex patterns, as it enables the preservation of hidden data details and their integration into the cover image without compromising quality. BN, meanwhile, is employed to enhance the stability and speed of the network during training. By normalizing data distribution in each layer, it assists the model in learning more efficiently and effectively. This ensures a more accurate placement and recovery of hidden data in the steganographic process. In addition to these primary modifications, adjustments in hyperparameters, like batch size and learning rate, along with the use of optimization algorithms and schedulers, as detailed in the datasets used and experimental setup section, have also served our main objective. These supplementary adjustments further refine the model's performance, ensuring that the steganographic process is both efficient and effective in embedding and recovering data without detection. The model commences with an initial residual block comprising 64 filters, each utilizing a 3×3 kernel. Internally, this residual block encompasses two convolutional layers, each featuring 64 filters with 3×3 kernels, followed by ReLU activation function and BN. This block operates on the input tensor, resulting in a feature map with 64 channels. Subsequently, a max pooling layer is applied with a 2×2 window and a stride of 2 to down-sample the features by half. The subsequent level introduces a new residual block containing 128 filters using 3×3 kernels. Similar to the previous block, this block consists of two convolutional layers with 128 filters each, followed by ReLU activation function and BN. After processing through this block, the feature map expands to 128 channels. Once again, a max pooling layer with a 2×2 window and a stride of 2 is employed to reduce the feature dimensions by half. This pattern continues throughout the encoder portion of the network, with each level doubling the number of filters in its respective residual block: 256 filters, and then 512 filters, followed by a corresponding max pooling layer. Finally, at the bottom level, the model employs a residual block featuring 1024 filters using 3×3 kernels.

The model then proceeds to initiate the expansion path. Upsampling is accomplished using transpose convolution, also known as deconvolution, employing a 3×3 kernel and a stride of 2, which effectively increases the size of the feature map by twofold. Following the upsampling step, a residual block is applied, reducing the number of filters by half, transitioning from 1024 to 512 filters. The output from the corresponding encoding level (512 filters) is concatenated with the output of this block. This pattern persists throughout the decoder section of the network, progressively reducing the number of filters in each residual block at each level: 256 filters, then 128 filters. Each of these steps is preceded by an upsampling operation and followed by a concatenation with the output from the relevant encoding level. In the decoder path, just like in the encoder path, each convolution process is followed by ReLU and BN in order to expedite network training. Final stage of model includes a 1×1 convolutional layer that produces a 3-channel output, essential for integrating the hidden data into the RGB channels during the steganography process. This is based on the fact that most digital images express color information through these three primary channels. Consequently, it is possible to efficiently encode the hidden data and then precisely extract it from the RGB channels.

The decoder, used to reveal the hidden image from the stego image, mirrors the encoder's architectural configuration, maintaining the same layer composition and corresponding filter dimensions. However, the operational dynamics are reversed; the decoder uses upsampling to expand the feature maps, in contrast to the encoder's downsampling process. This reversal is critical for the steganographic recovery process, enabling the decoder to rebuild the high-quality image from the hidden image. The proposed model's block diagram is displayed in Figure 2, and the architecture of the hiding network is illustrated in Figure 3. The efficacy of the suggested steganography method, with a 100% payload capacity, is evaluated using PSNR and SSIM metrics. PSNR assesses the amount of error and its impact on visual quality by evaluating the differences between the stego and the cover image. SSIM evaluates visual structural similarity and quality, enabling a comprehensive analysis of both the concealing and revealing processes' success and their effects on image quality. Additionally, histogram comparisons are also conducted for a more detailed evaluation of the performance of the model. This allows observation of changes in the color distributions of the images after the embedding process. Visual results are also presented, including comparative visuals of the cover, stego, message, and extracted message images. These measurement metrics and visual results enable the evaluation of our model's performance in steganography-specific parameters. These include criteria like imperceptibility, high hiding capacity, and reversibility. Imperceptibility ensures the hidden message remains undetectable, while high hiding capacity signifies the model's ability to embed substantial data without visible changes in the image. Reversibility allows for the accurate and complete extraction of the hidden message without any loss or distortion. Evaluated alongside PSNR, SSIM, and histogram comparisons, these metrics not only represent the outcomes but also form the fundamental components of our comprehensive evaluation strategy. This approach highlights the model's effectiveness in steganography and underscores its suitability for scenarios where data security is important.

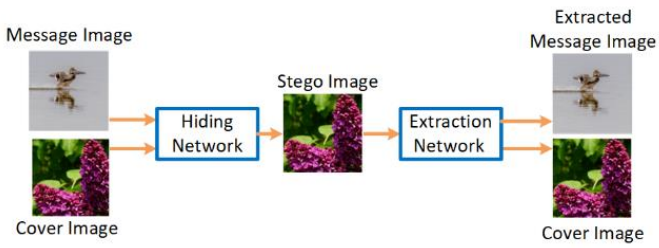


Figure 2. Block diagram of the suggested model

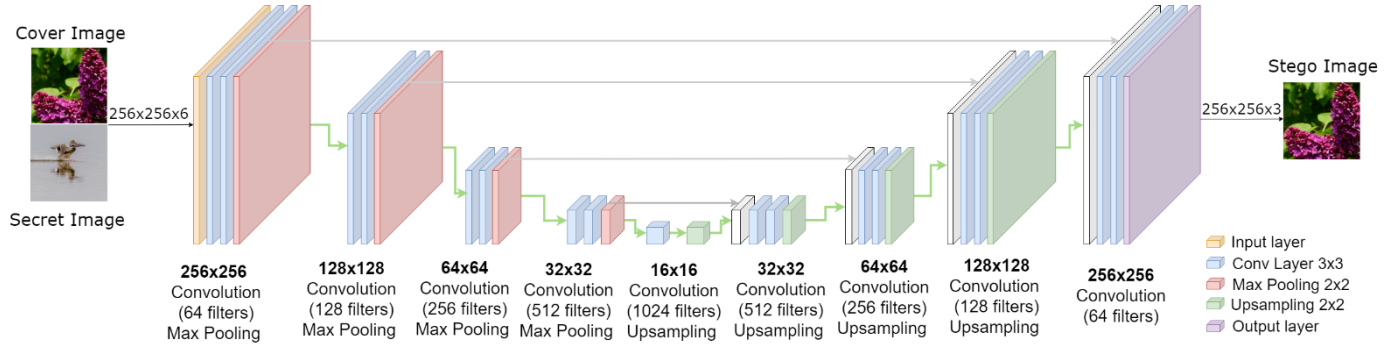


Figure 3. The architecture of hiding network

4. DATASETS USED AND EXPERIMENTAL SETUP

In our study, the training process is conducted using the ImageNet database. In the context, a random selection of 100.000 images is made from the ImageNet database and classified into two groups: cover and message images. Of these, 50.000 images designated as cover images, 37.500 are used for training and 7.500 for testing. In a similar manner, from the 50.000 images set aside as message images, 37.500 are allocated for training and 7.500 for testing. Before being fed into the model, the entire dataset is standardized to a consistent size of 256×256 pixels to match the model's input dimensions. Furthermore, normalization is applied to enhance the efficiency of the model's learning. This process involves scaling and standardizing each pixel value in the dataset to the [-1, 1] range, thereby optimizing the training process for efficiency and effectiveness.

In the validation phase of our model, the Linnaeus 5 dataset was employed. Designed for machine learning and image processing applications, Linnaeus 5 encompasses five primary classes: fruit, bird, dog, flower, and a category termed 'other'. The images are presented in various resolutions, including 256×256, 128×128, 64×64, and 32×32 pixels, and are colored. One rationale for utilizing this dataset in validation is its provision of a rich source in terms of visual diversity. Another reason is its inclusion of images with varying resolutions, which allows for an evaluation of the impact of hidden images of different original sizes on the cover image. Examples from the Linnaeus 5 dataset can be found in Figure 4.

As cover images, 256×256-pixel images of Linnaeus 5 dataset are utilized. Different-sized message images (32×32, 64×64, 128×128, 256×256) were chosen to examine the effect of this variation on the cover image. Before being fed into the model, hidden images are resized to 256×256 to fit into the model, and all cover and hidden images have been normalized to to the [-1, 1] range.

Figure 4. Examples of the Linnaeus 5 dataset

Our study's validation process, in addition to using the Linnaeus 5 database, has also been extended to include the ImageNet and LFW databases. The ImageNet database includes images of various resolutions sourced from real-world conditions, while the LFW database specifically houses face images with a resolution around 250×250 pixels, catering to face recognition studies. These two databases are widely referenced in the literature. This validation across multiple datasets is important for testing the generalization capabilities of our model and assessing how well it can adapt to the diversity of real-world conditions. This approach demonstrates that our model is not limited to a specific dataset but is also effective against different types and resolutions of data. It highlights the model's versatility and its ability to perform effectively across various datasets. For these datasets as well, the resizing and normalization steps were performed on the data before being fed into the model. Additionally, across three databases, the selection of cover and secret images is conducted from mutually exclusive subfolders to prevent any potential data leakage. This approach ensured that there was no overlap between the images in each database, thus preserving the uniqueness of each dataset.

The network has been initialized with a weight decay and learning rate, both of which are set at 0.001. The reason for choosing 0.001 as the initial learning rate is its balance between fast learning and generalization capability for the model. This value is a widely accepted standard in the literature and has been proven successful in various studies. It also helps in avoiding issues like overfitting and underfitting. Similarly, the weight decay value was carefully chosen for similar reasons. Other values were tested, but 0.001 was found to be the most suitable for our model.

In our study, we have employed the AdamW optimization method, which is renowned for its ability to automatically adjust the learning rate, thereby ensuring a smooth learning trajectory for network parameters. The choice of AdamW optimizer for the steganography task is particularly advantageous due to its enhancement of weight decay, which reduces the risk of overfitting, and its provision of adaptive learning rate adjustments. These features collectively enhance the model's capability to integrate and retrieve secret information within visual data, while also strengthening its generalization ability. In addition, the one cycle learning rate scheduler [43] has also been used to further improve the dynamic adjustment of the learning rate, hence enhancing the efficiency of achieving convergence to the minimum loss. This method starts with a minimum learning rate, increasing towards a maximum, enabling the model to explore a broad parameter space and avoid local minimum. After reaching the peak, the learning rate is systematically reduced, either

linearly or exponentially, allowing precise fine-tuning and reducing overfitting. We specifically implemented the one cycle learning rate scheduler in our project, starting with a learning rate of $1e^{-4}$ and gradually increasing to $1e^{-2}$, followed by a reduction to approximately $1e^{-6}$. The combination of AdamW's weight decay management and adaptive learning rate adjustments with the dynamic learning rate changes from the one cycle learning rate scheduler accelerates and enhances model training. This synergistic approach is effective in optimizing complex tasks like steganography, where the learning rate initially increases for extensive parameter space exploration and then decreases for fine-tuning around optimal parameters. A batch size of 32 per iteration has been chosen, and the network has been trained over 200 iterations. This choice targets an optimal balance between computational efficiency and the model's generalization capability. A batch size of 32 is sufficiently large to provide reliable gradient estimates at each step, yet it is not so large as to necessitate excessive computational resources or lead to poor generalization. Additionally, this size optimally utilizes the capabilities of the hardware used in our setup. The duration of 200 iterations has been determined through empirical testing, balancing the need for adequate learning and computational efficiency. This number of iterations has been identified as the optimal duration, ensuring sufficient model learning while preventing overfitting, making it particularly suitable for complex tasks such as steganography. Within the GPU, NVIDIA Tesla A100 40Gb is employed, the testing framework utilized is Pytorch 2.0.1+cu118, and Python 3.10 is used for conducting simulation experiments. Additionally, the libraries used in this study include: TensorFlow, torchvision, NumPy, Pandas, Matplotlib, PIL (Pillow), torchmetrics, skimage, pytorch_msssim, sklearn, plotly, os, re, copy, and pickle. Although optimized for high-performance GPUs, our model can also be trained on less powerful systems, though this may require some trade-offs. Training on hardware with lower computational capabilities might result in longer training periods and could demand changes in training parameters such as batch size and learning rate to suit the available resources.

5. EXPERIMENTAL RESULTS AND METRICS USED

In the optimization of our model during the training phase, a custom loss function is devised, which combines two fundamental components: MSE and SSIM, with the aim of establishing a balance between them. MSE measures pixel-level discrepancies between the model's output and the target, while SSIM is employed to provide a more encompassing perspective on structural similarity and image quality, assessing the structural and textural similarity between the model's output and the target. Our loss function is defined by a hyperparameter, denoted as α, which dictates the weight that is accorded to each component during combination. Mathematically, the total loss L is expressed as:

$$L = \alpha . MSE + (1 - \alpha) . (1 - SSIM) \quad (1)$$

The formula used to calculate the MSE value between the stego and the cover image is provided below.

$$MSE = \frac{\Sigma_{M,N} [I_1(m,n) - I_2(m,n)]^2}{M * N} \quad (2)$$

where, M and N are defined as the rows and columns of the images, respectively, in terms of pixel count. $I_1(m, n)$ depicts the pixel value at location (m, n) in the cover image. $I_2(m, n)$ symbolizes the pixel value at location (m, n) in the stego image. A smaller MSE denotes a higher similarity between the cover and the stego image, resulting in a decreased detectability of the steganography [44]. Using the same formula, the difference between the secret and the hidden image extracted from the cover image is computed. The loss graphs of the encoder-decoder model over epochs are shown in Figure 5.

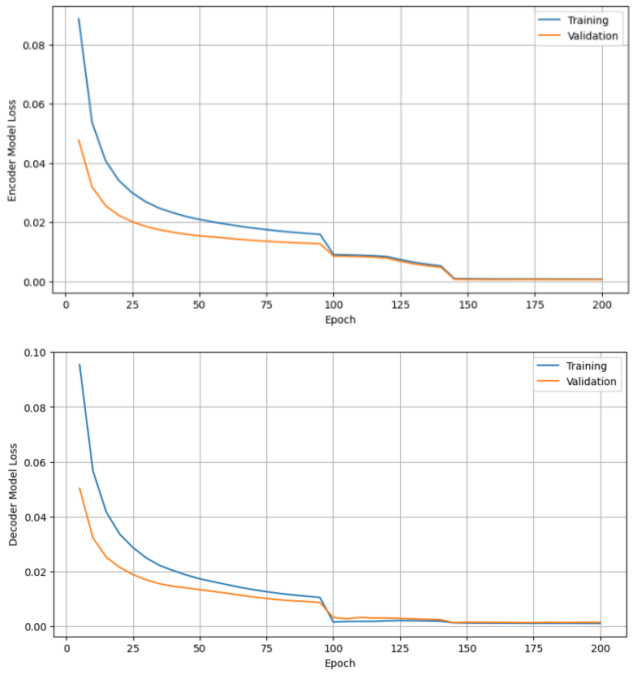


Figure 5. The loss graphs of the encoder-decoder model

In the scope of performance evaluation, PSNR and SSIM metrics are utilized to assess both the differences between the stego and the cover image, and the differences between the original and the extracted secret image. PSNR measures the average error rate per pixel between the images. This metric, especially when measuring the differences between the stego and the cover image, is utilized to evaluate how much error is generated by the embedding process and the impacts of these errors on visual quality. Similarly, the success of the extraction procedure is ascertained by measuring the differences between the extracted hidden and the original image. A greater PSNR value, measured in decibels, indicates that the secret image remains virtually undistorted when embedded within or extracted from the cover image [45]. PSNR is computed as follows:

$$PSNR = 10log_{10}(\frac{R^2}{MSE}) \quad (3)$$

where, R stands for the highest value a pixel can have in the image. It's calculated as $R=2^n-1$, where n represents the pixel depth. For an eight-bit image, this value is set at 255.

SSIM is a metric that assesses the structural resemblance between a pair of images. Unlike traditional methods which solely evaluate pixel-based differences, SSIM takes into

account structural information, contrast, and luminance changes.

Luminance similarity is defined as:

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad (4)$$

Contrast similarity is defined as:

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad (5)$$

Structure similarity is defined as:

$$s(x,y) = \frac{2\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \quad (6)$$

Combining these gives the full formula for SSIM:

$$SSIM = l(x,y) * c(x,y) * s(x,y) \quad (7)$$

where, μ_x and μ_y symbolize the mean of the images x and y, σ_x and σ_y represents standard deviations of images and $\sigma_x\sigma_y$ indicate the covariance between x and y. A value close to 1 ensures that the visual characteristics and patterns of the image remain consistent even after the embedding and extraction processes.

In the most challenging scenario, which involves hiding a same-size image inside a 256×256×3 cover image, the results indicate that the stego and the extracted message images have the same visual characteristics as the original cover and message images. The results of hiding and extracting 4 images are listed in Figure 6. The figure illustrates four columns: The first column contains the cover images, the second column displays the stego images, the third column shows the message images, and the fourth column depicts the extracted message images. Similarly, the visual results obtained from the ImageNet and LFW databases are presented in Figure 7 and Figure 8.

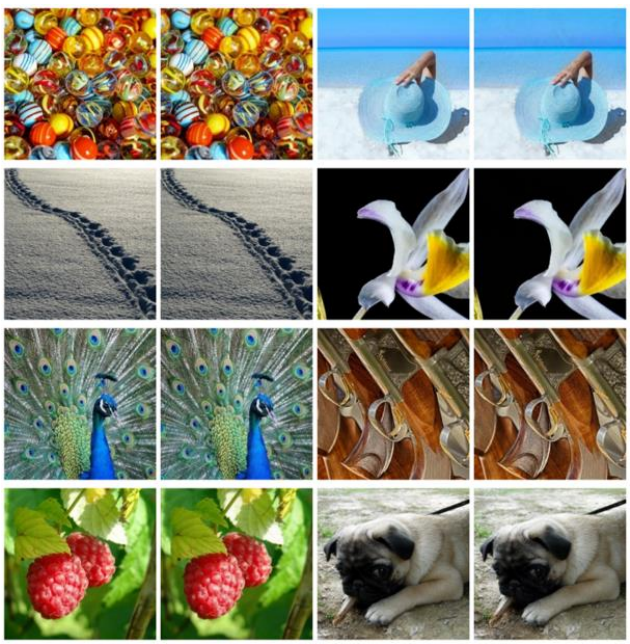


Figure 6. Visual comparison of hidden and extracted images for Linnaeus 5 dataset

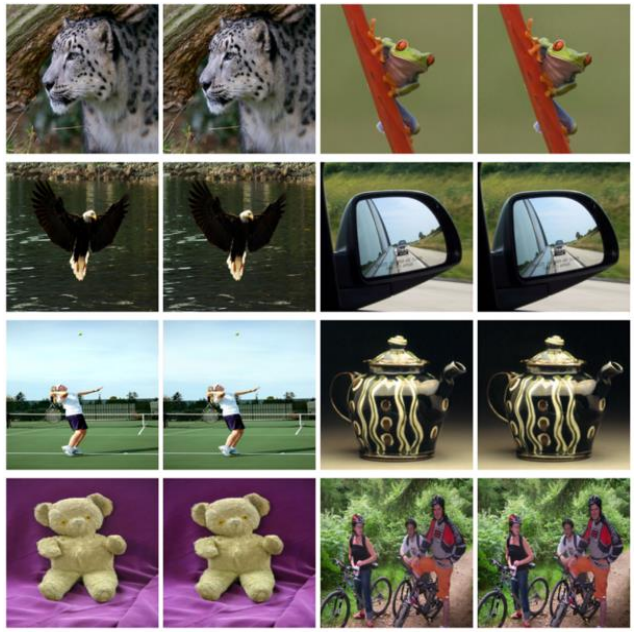


Figure 7. Visual comparison of hidden and extracted images for ImageNet dataset

Figure 8. Visual comparison of hidden and extracted images for LFW dataset

Table 1. PSNR and SSIM values for images in Figure 9

Stego-Cover PSNR (dB)	Extracted-Message PSNR (dB)	Stego-Cover SSIM	Extracted-Message SSIM
48.2361	49.2357	0.9959	0.9938

Table 2. PSNR and SSIM values for images in Figure 10

Stego-Cover PSNR (dB)	Extracted-Message PSNR (dB)	Stego-Cover SSIM	Extracted-Message SSIM
47.9897	47.7276	0.9929	0.9933

Table 3. PSNR and SSIM values for images in Figure 11

Stego-Cover PSNR (dB)	Extracted-Message PSNR (dB)	Stego-Cover SSIM	Extracted-Message SSIM
46.5265	46.3236	0.9904	0.9895

Table 4. PSNR and SSIM values for images in Figure 12

Stego-Cover PSNR (dB)	Extracted-Message PSNR (dB)	Stego-Cover SSIM	Extracted-Message SSIM
45.8864	45.5080	0.9856	0.9847

Table 5. PSNR and SSIM values for images in Figure 13

Stego-Cover PSNR (dB)	Extracted-Message PSNR (dB)	Stego-Cover SSIM	Extracted-Message SSIM
47.6706	45.6327	0.9867	0.9934

Table 6. PSNR and SSIM values for images in Figure 14

Stego-Cover PSNR (dB)	Extracted-Message PSNR (dB)	Stego-Cover SSIM	Extracted-Message SSIM
48.5492	48.0038	0.9914	0.9945

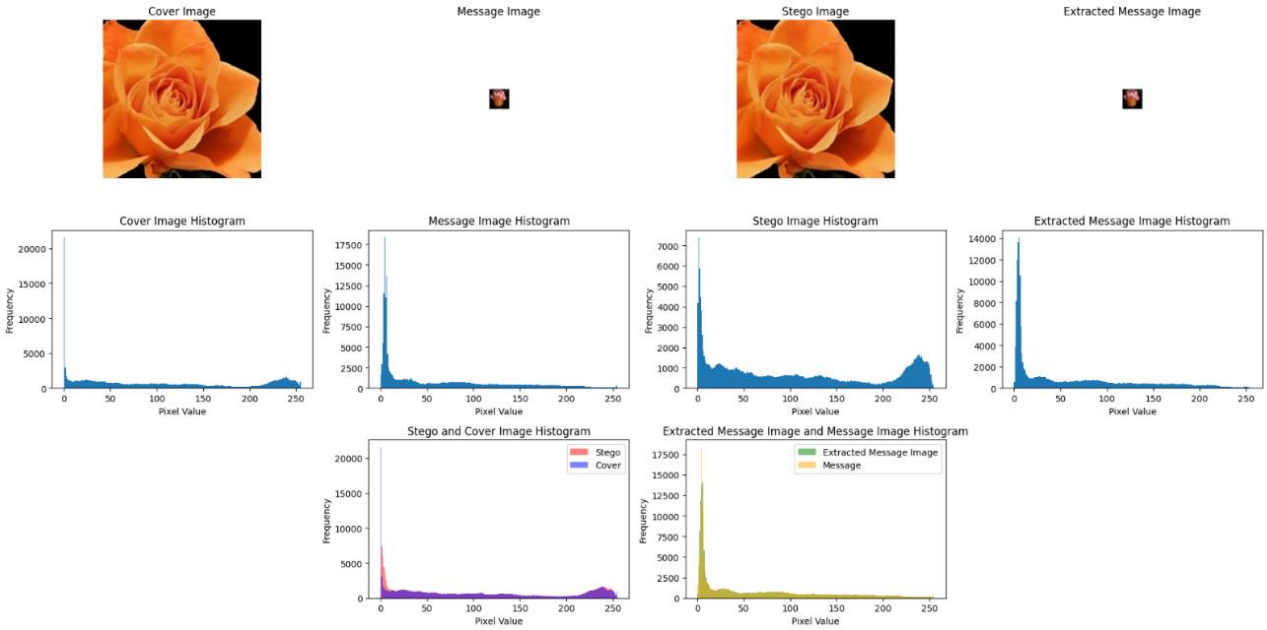


Figure 9. The distinction between cover image and message image before and after steganography in Linneaus 5 dataset (cover image 256×256×3, original message image 32×32×3)

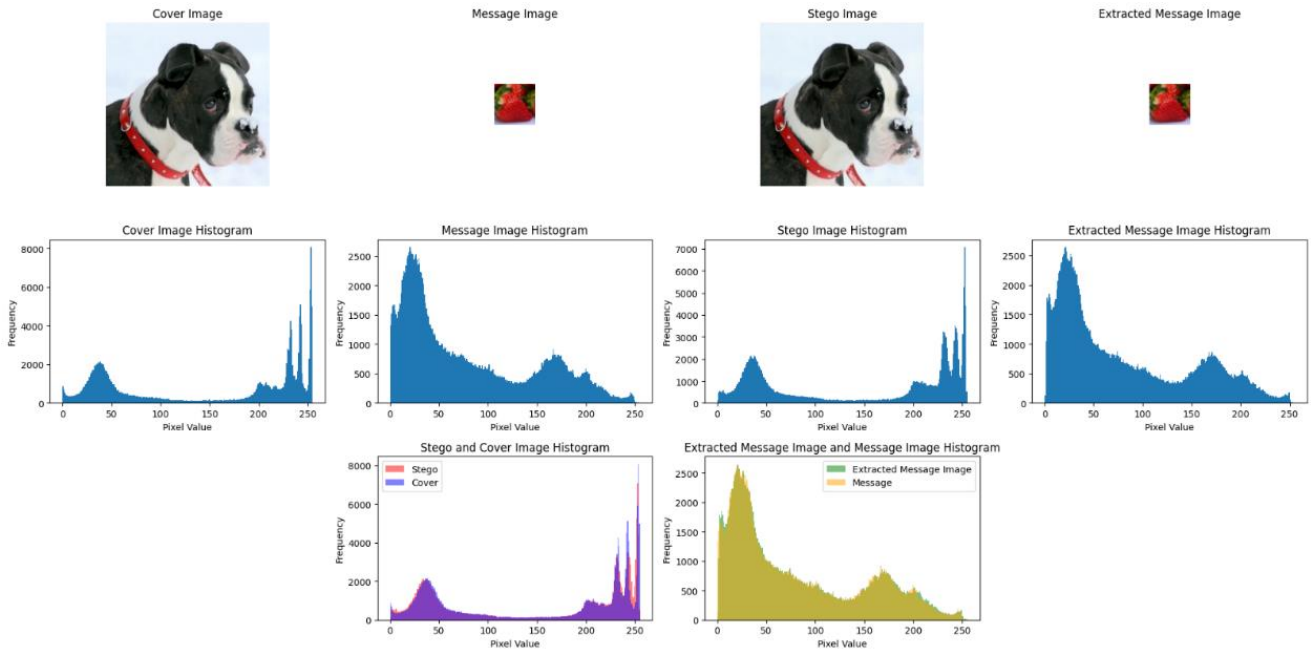


Figure 10. The distinction between cover image and message image before and after steganography in Linneaus 5 dataset (cover image 256×256×3, original message image 64×64×3)

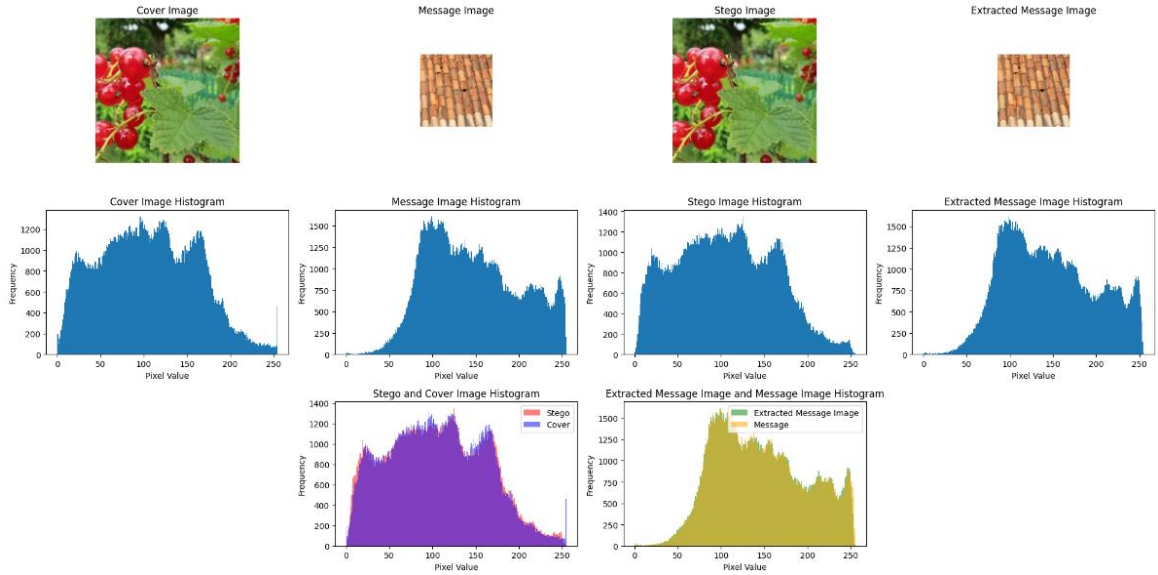


Figure 11. The distinction between cover image and message image before and after steganography in Linneaus 5 dataset (cover image 256×256×3, original message image 128×128×3)

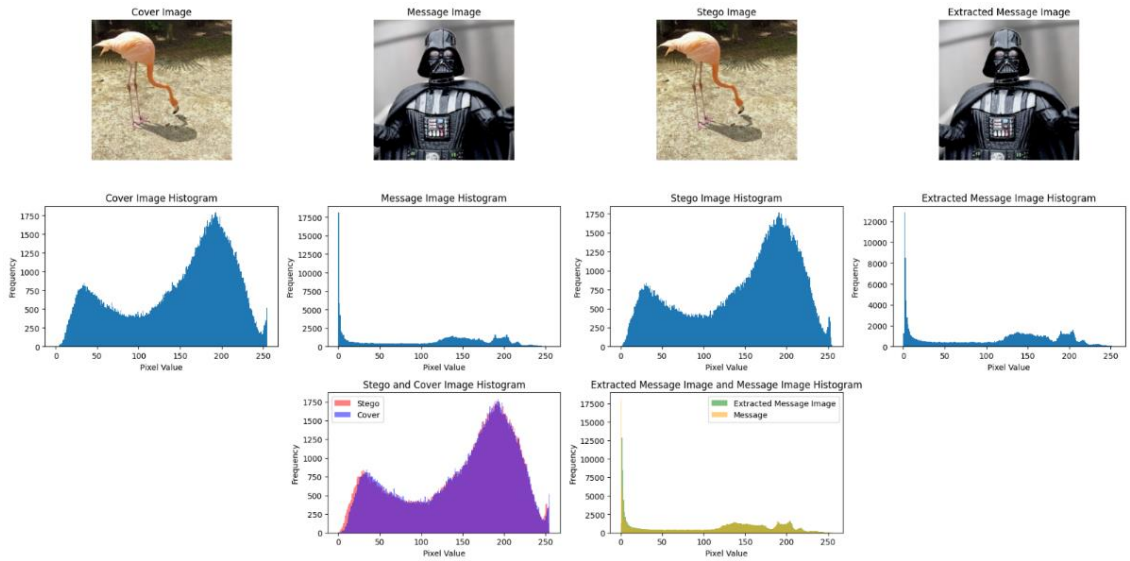


Figure 12. The distinction between cover image and message image before and after steganography in Linneaus 5 dataset (cover image 256×256×3, original message image 256×256×3)

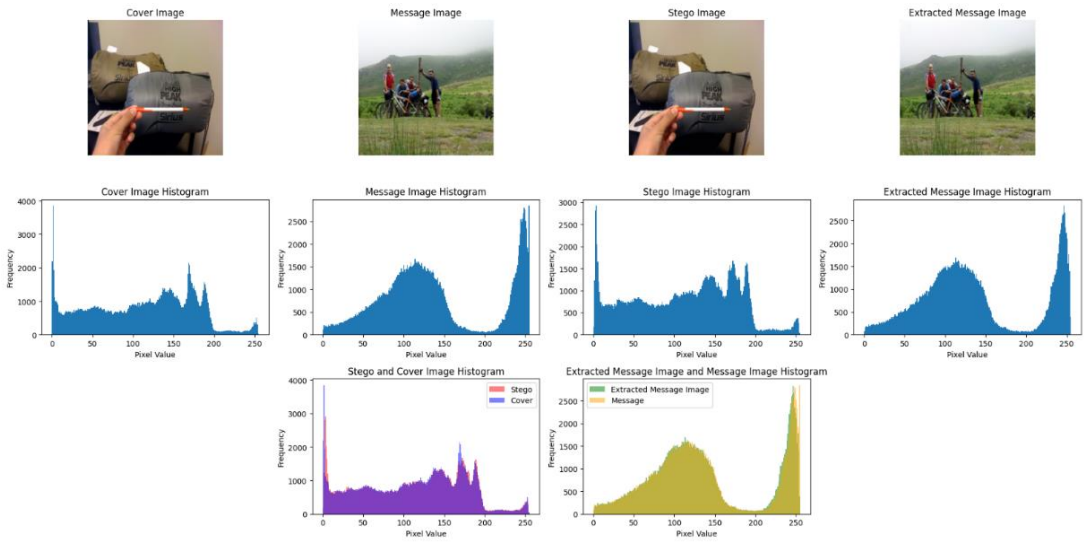


Figure 13. The distinction between cover image and message image before and after steganography in ImageNet dataset

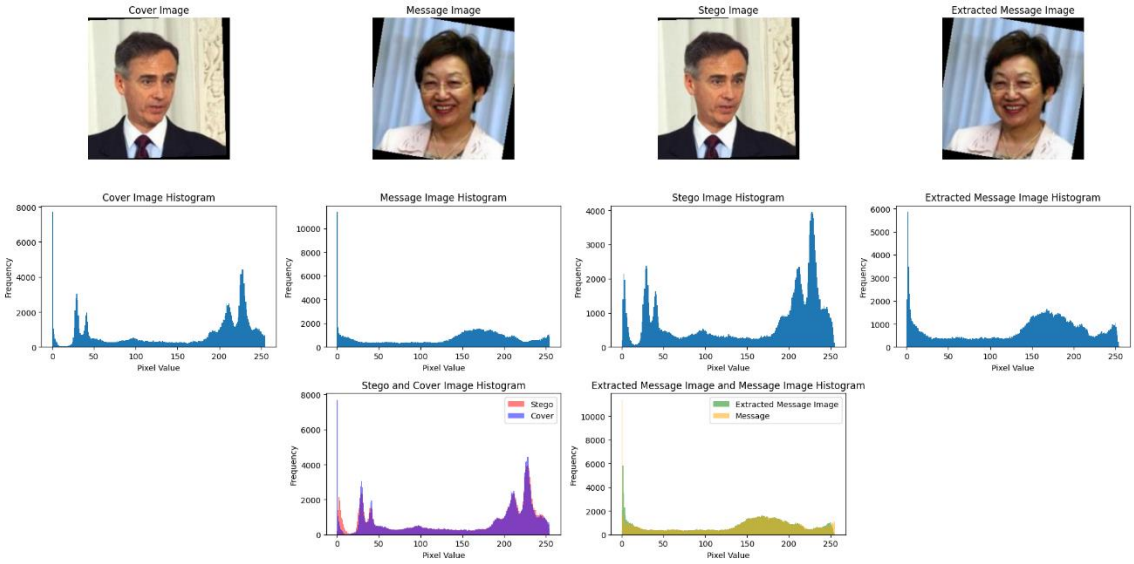


Figure 14. The distinction between cover image and message image before and after steganography in LFW dataset

The visual results and histogram graphs obtained from the concealment and extraction of colored message images with original sizes of 32×32, 64×64, 128×128, and 256×256 respectively, within a 256×256 sized colored cover image, are provided in Figures 9-12. As mentioned before, hidden images are resized to 256×256 to fit into the model, followed by a resizing back to their original resolution for plotting purposes The corresponding results, including the PSNR and SSIM values associated with these visual outcomes, are respectively detailed in Tables 1-4. Additionally, the mentioned results obtained from the ImageNet and LFW databases are presented in Figure 13 and Figure 14, and the metric results are included in Table 5 and Table 6.

Upon examining the average values presented in Table 7, it appears that when smaller original images with a size of 32×32 are resized to 256×256 and inputted into the architecture, the resulting PSNR and SSIM scores are at their highest. Our findings suggest that there is an inherent connection between the original message image sizes and the steganographic quality, as measured by PSNR and SSIM. This is due to the resizing and embedding process of the image, which affects the overall information density and consequently the steganographic outcome. Smaller images, when resized to a larger format, undergo interpolation which introduces a smoothing effect. This effect reduces high-frequency components and noise, which typically hinders steganography, and as a result, blends the image more seamlessly into the cover, yielding higher PSNR and SSIM values. On the contrary, larger original images carry more intricate details and higher informational content per pixel. When these images are embedded into the cover, they tend to introduce more noticeable alterations, due to both the increased amount of detail and the potential for more pronounced noise. These factors can disrupt the steganographic process, leading to lower PSNR and SSIM scores.

Table 7. Average PSNR and SSIM values for various original message image sizes

Cover Image Size	Original Message Image Size	Stego-Cover PSNR (dB)	Extracted-Message PSNR (dB)	Stego-Cover SSIM	Extracted-Message SSIM
256×256×3	32×32×3	49.5532	49.0738	0.9946	0.9940
256×256×3	64×64×3	49.0185	48.6563	0.9933	0.9924
256×256×3	128×128×3	47.9247	46.7569	0.9911	0.9902
256×256×3	256×256×3	44.4656	43.5393	0.9897	0.9875

In practical applications, the choice of original image size should be guided by the specific requirements of the steganographic task. For applications where maximum imperceptibility is critical, smaller original images may be preferred. However, for applications where the integrity of the embedded information is paramount, larger original images might be more appropriate despite the potential reduction in PSNR and SSIM metrics. However, it is important to emphasize in the context of our study that even when the cover and message images have the same dimensions, our results maintain a very high level of imperceptibility. This demonstrates the consistent performance of our model in the domain of steganography.

Overall, it is evident from the results of our study that we have achieved statistically significant and visually impressive outcomes. It is important to highlight that the effectiveness of our method lies in embedding supplementary information into every pixel in the cover image, rather than altering the pixel values directly. This subtle approach not only preserves the visual integrity of the cover image but also makes it challenging to detect the presence of the embedded data, showcasing the robustness of our steganography technique. Comparing the suggested approach's outcomes with those of other deep learning research, especially those involving U-Net, is shown in Table 8. In the referenced table, all studies except [39] utilized colored cover images of 256×256

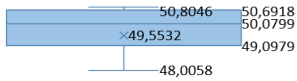
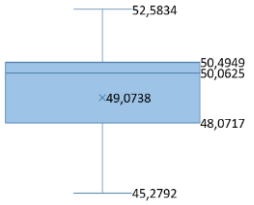
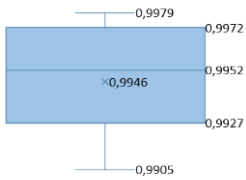
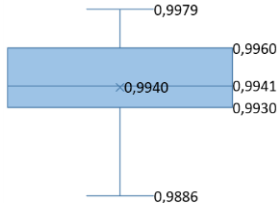
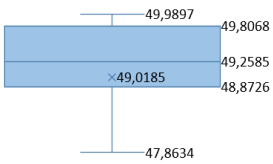
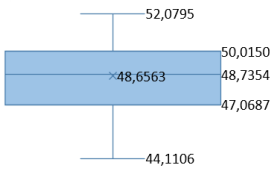
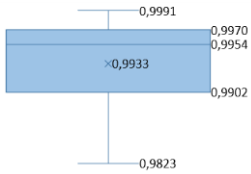
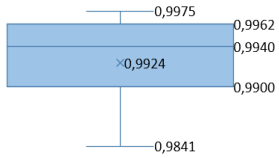
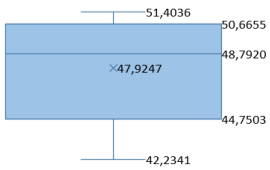
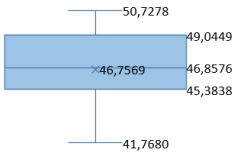
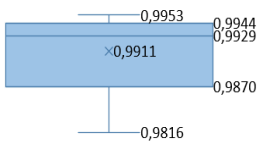
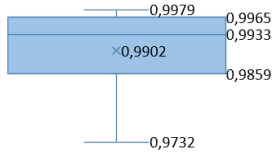
dimensions. Studies where the payload is indicated as 33% employed grayscale images of 256×256 dimensions as secret images. On the other hand, studies specifying a payload of 100% used colored images of 256×256 dimensions as message images. Only in the study [39], both cover and secret images of 224×224 dimensions in color were used. When the results are examined, it is observed that our study enhances the PSNR and SSIM values obtained in other studies which have same payload. This success reflects the impact of the methods employed. This clearly demonstrates the effectiveness of our approach in optimizing image quality in steganographic applications, achieving positive results in terms of statistical reliability and visual accuracy, thereby setting a benchmark for future research in this field.

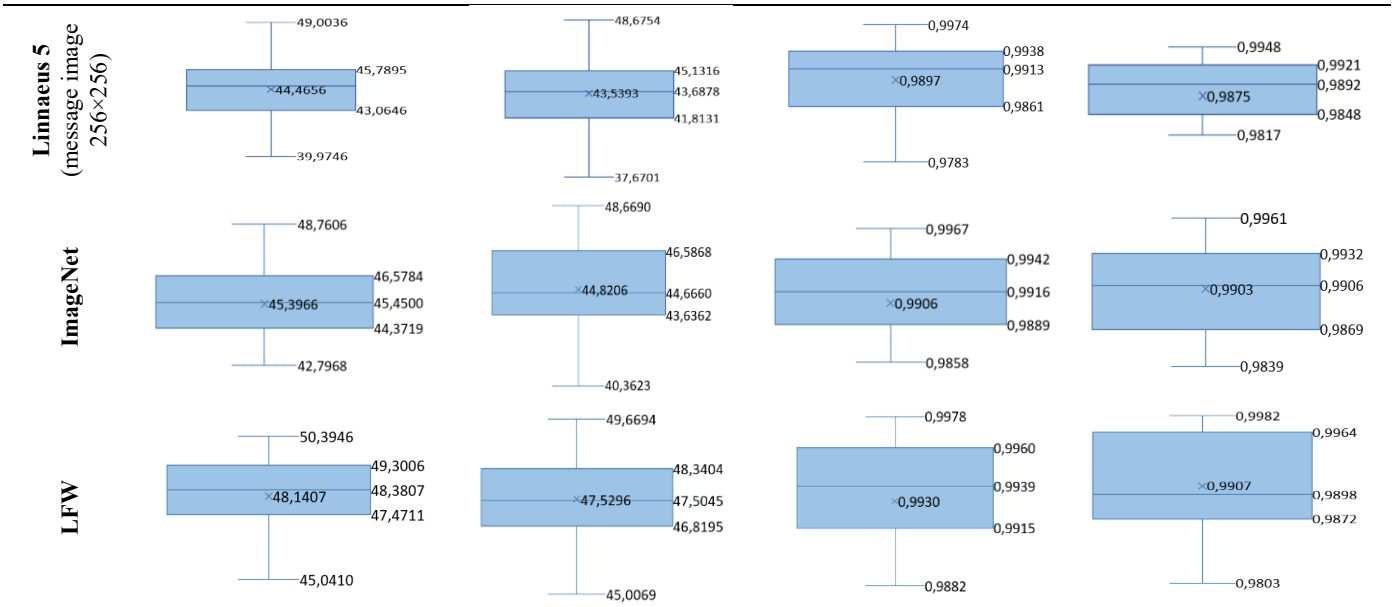
In Table 8, in addition to the results obtained from the Linnaeus 5 dataset, the average PSNR and SSIM results from the validation process of our model with the ImageNet and LFW databases are also presented. Upon examining these results, it is observed that the improvement in PSNR and SSIM levels continues and even increases. The notable quantitative and qualitative results obtained across three distinct datasets demonstrate the generalization ability of our model against various data types and conditions. It is evident that our model can successfully process diverse visual contents and complexities, effectively preserving steganographic secrecy. When evaluated in terms of application areas, the results clearly indicate our model's capacity to adapt to various image types encountered in real-world scenarios, including facial images, and to exhibit high steganographic performance in these situations.

Table 8. Result comparison between the suggested method and other deep learning approaches

Method	Technique	Payload	Stego-Cover PSNR (dB)	Extracted-Message PSNR (dB)	Stego-Cover SSIM	Extracted-Message SSIM
Rehman et al. [26]	CNN	%33	32.92	36.58	0.96	0.96
Baluja [27]	CNN	%100	41.2	37.6	0.98	0.97
Zhang et al. [31]	ISGAN	%33	34.57	36.58	0.9652	0.9465
Subramanian et al. [32]	Autoencoder	%100	34.55	27.93	-	-
Liu et al. [33]	U-Net	%33	39.7708	43.3571	0.9828	0.9862
Liu et al. [34]	U-Net	%33	40.8965	49.6028	0.9813	0.9963
Duan et al. [35]	U-Net	%100	40.4716	40.6665	0.9794	0.9842
Himthani et al. [24]	U-Net	%100	38.00	38.00	0.9875	0.9869
Himthani et al. [24]	V-Net	%100	30.00	33.00	0.9680	0.9810
Himthani et al. [24]	U-Net++	%100	24.00	27.00	0.910	0.930
Zeng et al. [36]	U-Net	%100	39.3912	35.8427	0.9894	0.9833
Wang [37]	U-Net++	%33	37.1381	35.4812	0.9768	0.9681
Wei et al. [38]	U-Net	%100	36.96	35.98	0.970	0.963
Jenynof & Ahmad [39]	U-Net	%100	28.539	29.759	-	-
Kich & Taouil [40]	U-Net	%100	37.83	31.77	0.9786	0.9077
Proposed (Linnaeus 5)	**U-Net**	**%100**	**44.4656**	**43.5393**	**0.9897**	**0.9875**
Proposed (ImageNet)	**U-Net**	**%100**	**45.3966**	**44.8206**	**0.9906**	**0.9903**
Proposed (LFW)	**U-Net**	**%100**	**48.1407**	**47.5296**	**0.9930**	**0.9907**

Table 9. PSNR and SSIM box graphs

Database	Stego-Cover PSNR	Extracted-Message PSNR	Stego-Cover SSIM	Extracted-Message SSIM
Linnaeus 5 (message image 32×32)				
Linnaeus 5 (message image 64×64)				
Linnaeus 5 (message image 128×128)				



In addition, the box plots in Table 9 offer a comprehensive statistical analysis of the PSNR and SSIM metrics for each investigated step. Each box plot features a central horizontal line representing the median, which serves as a visual indicator of the dataset's central value. The upper and lower boundaries of the box define the third and first quartiles (Q3 and Q1), respectively, encompassing the interquartile range (IQR) that reflects the middle 50% spread of the dataset. The 'whiskers' extend from the quartiles to the minimum and maximum values of the data, excluding any outliers, and map the full range of observed values. This comprehensive representation enhances the mean PSNR and SSIM values reported in Table 7 and Table 8 by incorporating the maximum and minimum values obtained, as well as the overall variance. Such a level of detail helps in understanding the consistency of the steganographic process's embedding and extraction quality across various datasets and conditions.

6. CONCLUSIONS

In our research, we developed a U-Net based steganography model specifically designed to embed and extract a message image of 256×256×3 dimensions into and from a cover image of the same size. Our model was trained using the ImageNet database. During the validation phase, two different analyses were conducted.

The first analysis involved examining the impact of secret images of various original sizes on the cover image during the steganography process. The second analysis evaluated the efficiency of our model across three different datasets. For the first analysis, the Linnaeus 5 database, which offers colored images of various sizes (32×32, 64×64, 128×128, 256×256) and a wide variety of image types, was utilized. Images smaller than 256×256 were resized to fit the architecture's input and were hidden inside colored cover images of 256×256 size for detailed analysis. For the 32×32×3 message image, the hiding phase achieved 49.5532 dB PSNR and 0.9946 SSIM, and the extraction phase achieved 49.0738 dB PSNR and 0.9940 SSIM. For the 64×64×3 size, the hiding phase recorded 49.0185 dB PSNR and 0.9933 SSIM, and the extraction phase recorded 48.6563 dB PSNR and 0.9924 SSIM. For the 128×128×3 size, the hiding phase achieved 47.9247 dB PSNR and 0.9911 SSIM, and the extraction phase achieved 46.7569 dB PSNR and 0.9902 SSIM. In the most challenging scenario, the 256×256×3 message image, our model reached 44.4656 dB PSNR and 0.9897 SSIM during the hiding phase, and 43.5393 dB PSNR and 0.9875 SSIM during the extraction phase. The results indicate that the quality of the extracted and stego images declines with increasing original image size, but high-quality levels are maintained across all sizes. Our findings demonstrate that our model provides a robust and effective solution for messages of varying sizes, offering potential applications in secure data transmission.

In the second analysis, our model's performance was also evaluated using the ImageNet and LFW datasets. This approach demonstrates the extent to which our findings may be generalised across different datasets. Additionally, it has provided the opportunity for direct comparison with other studies in the literature that have used the same datasets. For the ImageNet dataset, our model exhibited 45.3966 dB PSNR and 0.9906 SSIM during the hiding phase, and 44.8206 dB PSNR and 0.9903 SSIM during the extraction phase. On the LFW dataset, the model achieved 48.1407 dB PSNR and 0.9930 SSIM during the hiding phase, and 47.5296 dB PSNR and 0.9907 SSIM during the extraction phase. To the best of our knowledge, the results obtained from all datasets indicate promising outcomes compared to existing deep learning algorithms in the literature, particularly in terms of improvements in PSNR and SSIM metrics Furthermore, based on the results obtained from different datasets, our algorithm has proven to be robust and reliable against complex backgrounds and variations in object appearances, thereby being suitable for reliable image steganograph.

In our investigation, the classical U-Net architecture is adapted for steganography purposes by incorporating specific modifications like batch normalization and residual blocks. These enhancements enable the network to handle deeper structural training while efficiently embedding secret data into images and maintaining their undetectability without compromising on quality. In addition, the integration of the one cycle learning rate scheduler with the AdamW optimization algorithm enhances U-Net training, contributing to improved outcomes through faster and more stable convergence. This combination enhances the model's generalization capability while ensuring high accuracy and

reliability in learning, as indicated by the metric results.

For the statistical analysis of the obtained PSNR and SSIM values, box plots were utilized in our study. These plots visually display the distribution and central tendencies of these metrics, providing evidence for the statistical consistency of our results. Additionally, they substantiate the accuracy of our findings and the overall methodological robustness of our research.

In light of the aforementioned considerations, our study offers important improvements, particularly in the embedding and extraction of high-capacity hidden data in large-scale images. This contributes to a strong and effective solution in steganography, improving the state of the art as it exists now.

In our research, limitations were encountered in terms of hardware resources and computational intricacy. The successful outcomes of the study were significantly contributed to by the utilization of high-performance GPUs. Performance constraints might be anticipated in systems equipped with lower-end hardware. When the algorithm is trained on hardware with limited computational capacity, extended training times might be required, and adjustments in training parameters such as batch size and learning rate might be necessitated to align with available resources. Therefore, continuing efforts in optimization and the exploration of distributed computing techniques is important for ensuring the efficient functioning of the algorithm on hardware with limited resources. The implementation of these improvements is expected to broaden the applicability of the algorithm across diverse hardware environments and enhance its scalability.

The practical application areas of our research are quite extensive. It can be beneficial in various fields where data privacy and secure transmission are important, such as healthcare, social media, and cybersecurity.

Finally, we would like to highlight that, due to the nature of steganography, ethical considerations are important. Despite being a powerful tool for transmitting and preserving hidden data, the potential for misuse of steganography is high. For example, its use in illegal activities can cause serious concerns. Therefore, in presenting our research findings, we emphasize the importance of guidelines and policies that promote the ethical and responsible use of steganography.

REFERENCES

[1] Demirci, B. (2016). Comparison of image steganography methods and their performances. Graduate dissertation. Department of Information Technology Engineering, Selçuk University, Konya, Turkey.

[2] Channalli, S. (2009). Steganography: An art of hiding data. International Journal on Computer Science and Engineering, 1(3): 137-141. https://doi.org/10.48550/arXiv.0912.2319

[3] Beimel, A. (2011). Secret-sharing schemes: A survey. Coding and Cryptology. In Proceedings of the Third International Workshop (IWCC), Qingdao, China, pp. 11-46. https://doi.org/10.1F007/978-3-642-20901-7_2

[4] Shamir, A. (1979). How to share a secret. Communications of the ACM, 22(11): 612-613. https://doi.org/10.1145/359168.359176

[5] Simmons, G.J. (1984). The prisoners' problem and the subliminal channel. In: Chaum, D. (e.d.) Advances in Cryptology. Springer, Boston, MA. https://doi.org/10.1007/978-1-4684-4730-9_5

[6] Neeta, D., Snehal, K., Jacobs, D. (2006). Implementation of LSB steganography and its evaluation for various bits. In 2006 1st International Conference on Digital Information Management, Bangalore, India, pp. 173-178. https://doi.org/10.1109/icdim.2007.369349

[7] Nolkha, A., Kumar, S., Dhaka, V.S. (2020). Image steganography using LSB substitution: A comparative analysis on different color models. In Smart Systems and IoT: Innovations in Computing: Proceeding of SSIC 2019, pp. 711-718. https://doi.org/10.1007/978-981-13-8406-6_67

[8] Subhedar, M.S., Mankar, V.H. (2014). Current status and key issues in image steganography: A survey. Computer Science Review, 13-14: 95-113. https://doi.org/10.1016/j.cosrev.2014.09.001

[9] Mathkour, H., Al-Sadoon, B., Touir, A. (2008). A new image steganography technique. In 4th International Conference on Wireless Communications, Networking and Mobile Computing, Dalian, China, pp. 1-4. https://doi.org/10.1109/WiCom.2008.2918

[10] Altaay, A.A.J., Sahib, S.B., Zamani, M. (2012). An introduction to image steganography techniques. In 2012 International Conference on Advanced Computer Science Applications and Technologies (ACSAT), Kuala Lumpur, Malaysia, pp. 122-126. https://doi.org/10.1109/ACSAT.2012.25

[11] Abraham, A., Paprzycki, M. (2004). Significance of steganography on data security. In International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004, Las Vegas, NV, USA, pp. 347-351. https://doi.org/10.1109/ITCC.2004.1286660

[12] Kadhim, I.J., Premaratne, P., Vial, P.J., Halloran, B. (2019). Comprehensive survey of image steganography: Techniques, evaluations, and trends in future research. Neurocomputing, 335: 299-326. https://doi.org/10.1016/j.neucom.2018.06.075

[13] Mandal, P.C., Mukherjee, I., Paul, G., Chatterji, B. (2022). Digital image steganography: A literature survey. Information Sciences, 609: 1451-1488. https://doi.org/10.1016/j.ins.2022.07.120

[14] Chandramouli, R., Memon, N. (2001). Analysis of LSB based image steganography techniques. In Proceedings 2001 International Conference on Image Processing (Cat. No.01CH37205), Thessaloniki, Greece, pp. 1019-1022. https://doi.org/10.1109/icip.2001.958299

[15] Swain, G., Lenka, S.K. (2015). Pixel value differencing steganography using correlation of target pixel with neighboring pixels. In 2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), Coimbatore, India, pp. 1-6. https://doi.org/10.1109/icecct.2015.7226029

[16] Hussain, M., Wahab, A.W.A., Anuar, N.B., Salleh, R., Noor, R.M. (2015). Pixel value differencing steganography techniques: Analysis and open challenge. In 2015 IEEE International Conference on Consumer Electronics-Taiwan, Taipei, Taiwan, pp. 21-22. https://doi.org/10.1109/icce-tw.2015.7216859

[17] Kumar, V., Kumar, D. (2018). A modified DWT-based image steganography technique. Multimedia Tools and Applications, 77: 13279-13308. https://doi.org/10.1007/S11042-017-4947-8

[18] Patel, K., Ragha, L. (2015). Binary image steganography in wavelet domain. In 2015 International Conference on

Industrial Instrumentation and Control (ICIC), Pune, India, pp. 1635-1640. https://doi.org/10.1109/iic.2015.7151012
[19] Leng, H.S., Tsai, C.J., Wu, T.J. (2022). A multilayer steganographic method using improved exploiting modification directions scheme. IEEE Access, 10: 468-485. https://doi.org/10.1109/access.2021.3136883
[20] Wahab, O.F.A., Khalaf, A.A.M., Hussein, A.I., Hamed, H.F.A. (2021). Hiding data using efficient combination of RSA cryptography, and compression steganography techniques. IEEE Access, 9: 31805-31815. https://doi.org/10.1109/access.2021.3060317
[21] Roy, S., Islam, M.M. (2022). A hybrid secured approach combining LSB steganography and AES using mosaic images for ensuring data security. SN Computer Science, 3(2): 153. https://doi.org/10.1007/s42979-022-01046-8
[22] Zhou, X., Gong, W., Fu, W., Jin, L. (2016). An improved method for LSB based color image steganography combined with cryptography. In 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), Okayama, Japan, pp. 1-4. https://doi.org/10.1109/ICIS.2016.7550955
[23] Patel, P., Patel, Y. (2015). Secure and authentic DCT image steganography through DWT-SVD based digital watermarking with RSA encryption. In 2015 Fifth International Conference on Communication Systems and Network Technologies, Gwalior, India, pp. 736-739. https://doi.org/10.1109/csnt.2015.193
[24] Himthani, V., Dhaka, V.S., Kaur, M., Rani, G., Oza, M., Lee, H.N. (2022). Comparative performance assessment of deep learning based image steganography techniques. Scientific Reports, 12(1): 16895. https://doi.org/10.1038/s41598-022-17362-1
[25] Subramanian, N., Elharrouss, O., Al-Maadeed, S., Bouridane, A. (2021). Image steganography: A review of the recent advances. IEEE Access, 9: 23409-23423. https://doi.org/10.1109/access.2021.3053998
[26] Rehman, A.U., Rahim, R., Nadeem, M.S., Hussain, S.U. (2019). End-to-end trained CNN encoder-decoder networks for image steganography. In: Leal-Taixé, L., Roth, S. (eds) Computer Vision – ECCV 2018 Workshops. Lecture Notes in Computer Science, Springer, Cham. https://doi.org/10.1007/978-3-030-11018-5_64
[27] Baluja, S. (2020). Hiding images within images. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(7): 1685-1697. https://doi.org/10.1109/tpami.2019.2901877
[28] Wu, P., Yang, Y., Li, X. (2018). Image-into-image steganography using deep convolutional network. In: Hong, R., Cheng, W.H., Yamasaki, T., Wang, M., Ngo, C.W. (eds) Advances in Multimedia Information Processing – PCM 2018. Lecture Notes in Computer Science, Springer, Cham. https://doi.org/10.1007/978-3-030-00767-6_73
[29] Wu, P., Yang, Y., Li, X. (2018). Stegnet: Mega image steganography capacity with deep convolutional network. Future Internet, 10(6): 54. https://doi.org/10.3390/fi10060054
[30] Yang, K., Chen, K., Zhang, W., Yu, N. (2019). Provably secure generative steganography based on autoregressive model. In: Yoo, C., Shi, Y.Q., Kim, H., Piva, A., Kim, G. (eds) Digital Forensics and Watermarking. IWDW 2018. Lecture Notes in Computer Science, Springer, Cham. https://doi.org/10.1007/978-3-030-11389-6_5
[31] Zhang, R., Dong, S., Liu, J. (2019). Invisible steganography via generative adversarial networks. Multimedia Tools and Applications, 78: 8559-8575. https://doi.org/10.1007/s11042-018-6951-z
[32] Subramanian, N., Cheheb, I., Elharrouss, O., Al-Maadeed, S., Bouridane A. (2021). End-to-end image steganography using deep donvolutional autoencoders. IEEE Access, 9: 135585-135593. https://doi.org/10.1109/ACCESS.2021.3113953
[33] Liu, L., Meng, L., Peng, Y., Wang, X. (2021). A data hiding scheme based on U-Net and wavelet transform. Knowledge-Based Systems, 223: 107022. https://doi.org/10.1016/j.knosys.2021.107022
[34] Liu, L., Meng, L., Zheng, W., Peng, Y., Wang, X. (2022). A novel high-capacity information hiding scheme based on improved U-Net. Security and Communication Networks, 2022: 1-12. https://doi.org/10.1155/2022/4345494
[35] Duan, X., Jia, K., Li, B., Guo, D., Zhang, E., Qin, C. (2019). Reversible image steganography scheme based on a U-Net structure. IEEE Access, 7: 9314-9323. https://doi.org/10.1109/access.2019.2891247
[36] Zeng, L., Yang, N., Li, X., Chen, A., Jing, H., Zhang, J. (2023). Advanced image steganography using a U-Net based architecture with multi-scale fusion and perceptual loss. Electronics, 12(18): 3808. https://doi.org/10.3390/electronics12183808
[37] Wang, Z. (2022). End-to-end image steganography scheme based on U-Net++ Structure. In 2022 4th International Conference on Frontiers Technology of Information and Computer (ICFTIC), Qingdao, China, pp. 1-6. https://doi.org/10.1109/ICFTIC57696.2022.10075116
[38] Wei, B., Duan, X., Nam, H. (2022). Image steganography with deep learning networks. In 2022 13th International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Republic of Korea, pp. 1371-1374. https://doi.org/10.1109/ICTC55196.2022.9952432
[39] Jenynof, A., Ahmad, T. (2023). Image to image steganography using U-Net architecture with mobilenet convolutional neural network. In 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), Delhi, India, pp. 1-7. https://doi.org/10.1109/ICCCNT56998.2023.10306352
[40] Kich, I., Taouil, Y. (2021). CNN auto-encoder network using dilated inception for image steganography. International Journal of Fuzzy Logic and Intelligent Systems, 21(4): 358-368. https://doi.org/10.5391/ijfis.2021.21.4.358
[41] Ronneberger, O., Fischer, P., Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W., Frangi, A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Lecture Notes in Computer Science, Springer, Cham. https://doi.org/10.1007/978-3-319-24574-4_28
[42] Siddique, N., Sidike, P., Elkin, C., Devabhaktuni, V. (2021). U-Net and its variants for medical image segmentation: A review of theory and applications. IEEE Journals & Magazine, 9: 82031-82057.

https://doi.org/10.1109/access.2021.3086020
[43] Smith, L.N., Topin, N. (2019). Super-convergence: Very fast training of neural networks using large learning rates. arXiv, 11006: 369-386. https://doi.org/10.48550/arXiv.1708.07120
[44] Das, P., Ray, S., Das, A. (2017). An efficient embedding technique in image steganography using Lucas sequence. International Journal of Image, Graphics and Signal Processing, 9(9): 51-58. https://doi.org/10.5815/ijigsp.2017.09.06
[45] Setiadi, D.R.I.M. (2021). PSNR vs SSIM: Imperceptibility quality assessment for image steganography. Multimedia Tools and Applications, 80: 8423-8444. https://doi.org/10.1007/s11042-020-10035-z