ISSN: 0067-2904

Enhancing Steganography System Using Deep Learning and Hyperparameter Optimization

Noor Redha Alkazaz[1,2], Asraa Muayed Abdalah[1]*
[1]*Department of Computer Science, College of Science for Women, University of Baghdad, Baghdad, Iraq.*
[2]*Biomedical Applications Department, College of Artificial Intelligence, University of Baghdad, Baghdad, Iraq.*

Received: 15/9/2024 Accepted: 2/7/2025 Published: 30/6/2026

Abstract:

Steganography plays a very important role in secure message transmission, embedding information imperceptibly within cover images. This research addresses the challenge of creating stego images visually indistinguishable from originals, while enhancing resistance to steganalysis. We propose a novel steganographic system based on Generative Adversarial Networks (GANs) with hyperparameter tuning to enhance imperceptibility and security. Multi-scale feature extraction via Inception modules within the encoder facilitates efficient embedding with minimal distortion. A new combined metric, incorporating MSE, PAD, and MAE alongside PSNR and SSIM, provides a more comprehensive performance evaluation, addressing limitations of relying solely on PSNR/SSIM. Multiple optimization algorithms, including Adadelta, Adam, AdamW, SGD, SparseAdam, Adamp, RAdamplus, Tadam, RMSprop, Nadam, Ftrl, AdaGrad, Nosadam, Adamwt, RAdam, L-BFGS, and Naturalgrad, were evaluated. Nadam and Adadelta achieved the highest combined metric scores (36.71 and 34.61, respectively), correlating with superior PSNR and SSIM values, thus ensuring higher image fidelity and imperceptibility. The system effectively impedes steganalysis by reducing statistical discrepancies and utilizes tanh activation functions for enhanced security. This system achieves robust steganographic performance with minimal distortion and improved security, offering significant advancements in secure image communication.

Keywords: Steganography, Deep Learning, GAN Network, Residual Module, Inception Module.

تعزيز نظام الإخفاء باستعمال التعلم العميق وتحسين المعلمات الفائقة

نور رضا القزاز[1,2]، أسراء مؤيد عبد الله[1*]

[1]قسم علوم الحاسوب، كلية العلوم للبنات، جامعة بغداد، بغداد، العراق

[2]قسم التطبيقات الطبية الحيوية، كلية الذكاء الاصطناعي، جامعة بغداد، بغداد، العراق

الخلاصه:

تُعدّ تقنية إخفاء المعلومات (Steganography) ذات أهمية بالغة في مجال الاتصالات الآمنة، حيث يتم تضمين المعلومات بشكل غير محسوس داخل الصور المُستخدمة كغطاء. يتناول هذا البحث تحدّي إنشاء

*Email: asraa.m@csw.uobaghdad.edu.iq

صور مُضمّنة (stego images) لا يمكن تمييزها بصريًا عن الصور الأصلية، مع تعزيز مقاومة النظام لتقنيات التحليل الإخفائي (steganalysis) المُستخدمة للكشف عن البيانات المخفية. نقترح نظامًا جديدًا لإخفاء المعلومات يعتمد على الشبكات التوليدية الخصومية (GANs) مع ضبط المعلمات الفائقة (hyperparameter tuning) لتعزيز خاصيتيّ عدم القابلية للكشف والأمان. يُسهل استخلاص الميزات متعددة المقاييس عبر وحدات Inception داخل المُشفِّر (encoder) عملية التضمين الفعّالة مع الحد الأدنى من التشوه. توفر مقياس مُركّب جديد، يتضمن MSE و PAD و MAE بالإضافة إلى PSNR و SSIM، تقييمًا أكثر شمولية للأداء، ويعالج أوجه القصور في الاعتماد فقط على PSNR / SSIM. على سبيل المثال، بينما حقق RMSprop قيمة PSNR تبلغ 57.32 ديسيبل، كانت درجة المقياس المركب الخاص به 16.37، مما يعكس نقاط ضعفه في مقاييس الخطأ. في المقابل، حقق SGD، بقيمة PSNR تبلغ 59.55 ديسيبل، درجة مقياس مُركّب قدرها 34.73، مما يدل على أدائه المتوازن. تم تقييم خوارزميات تحسين مُتعدّدة، بما في ذلك Adadelta و Adam و Nosadam و SparseAdam و Adamp و L-BFGS و RAdamplus و Tadam و RMSprop و Nadam و Ftrl و AdaGrad و AdamW و Adamwt و SGD و RAdam و Naturalgrad. حقق Nadam و Adadelta أعلى درجات في المقياس المركب (36.71 و 34.61 على التوالي)، مما يتوافق مع قيم PSNR و SSIM المُتفوقة، وبالتالي ضمان جودة صورة أعلى وعدم قابلية للكشف بشكل أفضل. يُعيق النظام بشكل فعال التحليل الإخفائي عن طريق تقليل التباينات الإحصائية ويستخدم دوال التنشيط tanh لتعزيز الأمان. يُحقق هذا النظام أداءً قويًا في إخفاء المعلومات مع الحد الأدنى من التشوه وتحسين الأمان، مما يُقدم تطورات كبيرة في مجال الاتصالات الآمنة القائمة على الصور. يُوفر المقياس المُركّب الجديد تقييمًا دقيقًا لأداء النظام، ويُرشد إلى اختيار الخوارزميات المُثلى للتطبيقات العملية.

1. Introduction

Steganography is an important method for hiding confidential information inside innocuous carrier medias. Modern steganographic algorithms use Deep Learning Algorims which intend to embed the maximum amount of information within an image with minimal visible changes on the image; such algorithms like Convolutional Neural Networks (CNNs) [1].

Gated Recurrent Unit (GRU), Long-short Term Memory (LSTM) Networks, Residual Networks (ResNet), Graph Neural Networks (GNNs), Convolutional Neural Networks (CNN), Capsule Networks (CapsNet) [2], Recurrent Neural Networks (RNN), Feed Forward Deep Networks (FDN), Multi-layer Belief Networks (DBN), and Adversarial-trained Generative Networks (AGN), are a different types of deep learning architectures or structures [3]. Figure 1 illustrates the learning workflow for a typical artificial intelligence system.

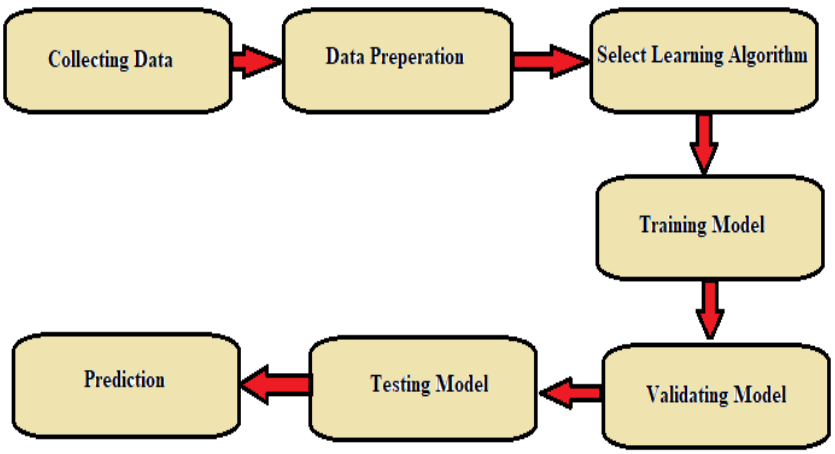

Figure 1: The Work of Machine Learning.

Generative Adversarial Networks (GANs) considered to be specialized AI models designed for generative modeling. These networks learn the distribution of data from image datasets and then produce new, realistic samples that follow the same distribution. These networks are able to produce high-resolution, photo-realistic images and this has made them a strong candidate for image-based security applications. In recent years, GAN-based deep learning models have shown promising potential for improving steganography techniques [4].

In Deep Neural Network (DNN); Hyperparameters are the made settings before training process. In contrast with weights and bias which considered the model's inherent parameters; hyperparameters are determined by the user's sense and evaluation, not on information learned from training dataset[5]. The correct selection of hyperparameters leads to great impact on the performance of neural networks and their ability to extract patterns and knowledge from the dataset and making generalized inferences. Thus hyperparameter optimization is essential for improving the performance of the Deep Neural Networks (DNNs) because it affects learning efficiency and model generalization [6]. Key hyperparameter include number of neurons and hidden layers, number of epochs, learning rate, activation fuction, batch size, and the optimizer type. To find optimal configuration, methods such as grid search, random search and Bayesian optimization are used [7].

Improving the performance in the steganographic cover images by using an Inception Module into a GAN-based framework via hyperparameter tuning is the main aim of the suggested network architecture. So the network's architecture design and performance assessment of the networks' performance on steganography tasks are discussed in this research. This research focus on building an enhanced steganography scheme by combining these two methods to utilize GANs' and Inception Modules' advantages through exploiting their strengths towards achieving efficient embedding of sensitive data while preserving visual integrity in cover materials.
Key contributions in this research can be summarized as follows:
1- Novelty of Network Architecture.
2- Hyperparameter Tuning.
3- Comprehensive Performance Evaluation.

Contributions will be explained in detail in the methodology section of this paper. Related work and highlights of the existing challenges in steganographic research are presented in Section 2. Section 3 details the proposed methodology and main contributions. Section 4 shows the proposed network architecture. Section 5 details performance evaluation. Section 6 discusses results, while Section 7 details conclusions and future work.

2. Related Work

There are many examples of the use of Artificial Intelligence for defensive measures for security, such as the research of Christian Tommel's [8] focus on the use of AI in offensive security. The proposed system by Kim et al., which is a Machine Learning-driven intrusion detection system (IDS) [9,10]. also Pattanayak and Ludwig [11] proposed a neural encryption system. Also, there are many review studies on steganography. The describtion of the weak and strong features of the important deep learning based steganography algorithms is shown in M. Arif and Bisma Sultan review paper [12]. Also a review described the use of deep learning for reversible image steganography proposed by Jyoti Khandelwal and VijayKumar [13]. In a review paper proposed by Hamza Kheddar et al. [14], they enhanced the performance of the steganalysis system by using deep transfer learning (DTL) and deep reinforcement learning (DRL). Anurag Singh et al. [15] in their survey, answered this

question (how can we efficiently hide more than one image in a single image and maintain the quality of the secret and cover images?).

2.1 Traditional image processing methods

An approach proposed by Hamza et al. [16] relayed on cryptographic techniques and traditional image processing. They used discrete shearlet transform and secret sharing schemes. Their Peak Signal to Noise Ratio (PSNR) achived 54.15 and this means the effectiveness of their method in preserving image quality.

Mithal Hadi et al. [17] proposed a steganographic approach that utilizes moving objects within video sequences to hide secret images. The method first isolates the moving object from the background to ensure accurate embedding. Then the secret image is positioned within the selected object based on its size and spatial characteristics. During the embedding phase the bits of the secret image are separated and combined with the bits of the detected moving object using an XOR-based bit-wise operation. In some cases, to increase the security; the bit order is reversed before embedding[18]. Unlike traditional video steganography methods that depend on static background regions for data hiding, this approach use dynamic moving objects as carriers. This improves security and imperceptibility because human visual perception is less sensetive to subtle changes in the moving regions compared to static regions. Experimental results showed the effectiveness of the method in achieving high visual quality in the stego video with a PSNR of 58.45 dB including minimal distortion and high fidelity.
Yusuf ŞANLI et al. [19] used Least Significant Bit (LSB) and Pseudo-Random Number Generator (PRNG) to randomly hide data in pixel bits to improve confidentiality. Apichat Heednacram et al. [20] distributed hidden data across the image by using Discrete Cosine Transform (DCT) and addaptive techniques to get better security and image qualty.

Ant Colony Optimization (ACO) with Discrete Cosine Transforms (DCT) is used by Ahmed Shihab et al. [21] to improve the robustness of steganography by selecting the best coefficients or pixel locations to hide data, so it becomes more secure and less detectable. Also least-significant-bits (LSB) substitution and ant colony optimization are used by Mariusz Boryczka and Grzegorz Kazana [22] for steganography.

2.2 Deep learning methods:

Improving visibility, capacity, and higher resistance to cryptanalysis attacks was the main objective of the work by Eman S. et al. [23] the used CNN to develop a Steganography techniques for videos. They performed image preprocessing methods to increase output quality and block-shuffling encryption to further strengthen security. Also they enhanced information hiding and security by arranging CNN architecture and weights randomly. Their training dataset size 45,000 including randomly selected colored images from the ImageNet dataset using a TensorFlow backend and Keras implementation.

2.3 GAN methods:

To enhance detection and resembles real-world patterns especially when datasets are small or need enhancement; Generative models such as Autoregressive Models, Generative Adversarial Networks (GANs), Transformer-based Generative Models and Variational Autoencoders (VAEs) are used to produce high-quality synthetic data that looks similar to real data. However, these models suffer from low interpretability [24] and high computational cost. For example Zhiwu Chen et al. [25] used GANs to enhance small dataset. While Francesco Mercaldo et al. [27] focused on detecting facke GAN-generated images using retina images dataset.

Ian Goodfellow et al. [28] proposed an overview study. They discussed the Generative Adversarial Networks which is a specific method of generative modeling for unsupervised learning. Their overview called for resolving the fundamental research issues around game convergence before GANs can be considered a dependable technology. Also it included a brief overview of GAN applications.

Using Deep Convolutional Generative Adversarial Networks (DCGANs), Sindhura et al. [29] proposed an effective approach for addressing the challenge of sub-axial spine fracture recognition. Their method focuses on generating high quality and realistic CT scan images of fractured spines, which helps address the chaleenge of the imbalanced datasets. The model imroves its ability to learn different features by augmenting the limited number of abnormal cases with synthetically generated samples, this enhancing the accuracy and robustness of fracture detection systems.

Faizan Munawar et al. [30] supports automated and reliable lung segmentation method, which can aid clinicians in faster and more accurate chest X-ray analysis. integrated four distinct discriminators with the GAN module architecture to improve model's precision in isolating lung areas from other thoracic structures in radio graphic images.

Maha Mesfer et al. [31] highlighted that Loey et al. [32] had proposed a generative adversarial network (GAN) model.

GAN used to generate realistic faces by training both the generator and the discriminator on image datasets. Fake faces were created by the generator; while the discriminator learned to detect them. This method has been widely studied and analyzed by Momina Masood et al. [33].

Researchers used GAN to advance steganography; Volkhonskiy et al. [34] explored manipulating cover images to bypass detection mechanisms, while Qin et al. [35] introduced a coverless strategy that encodes messages directly via generative networks. Building on this, Yao et al. [36] fused DWT with GANs to boost imperceptibility, and Wang et al. [37] combined genetic algorithms with GANs to constrain embedding regions, effectively preserving visual fidelity.

3. Methodology

The approach used in this paper comprises the following steps:

3.1 Dataset Preparation: Atotal of 46507 Images were collected from 17 Arabic movies and were used as cover and hidden images in the steganography system. This dataset includes a mix of images types: some images are black and white images taken from older movies, while others are colored images taken from more recent movies. To benefit from the cultural and regional diversity inherent in Arabic movies; the dataset images were taken from these movies instead of using standard benchmark datasets. To improve protection and effectiveness of steganography method; Arabic films were used because it contain rich, complex and diverse visual characteristics such as illuminations, facial reactions, and environmental details, which are essential for determining the robustness of the steganography system in real-life conditions. Images taken from old movies are black and white and have low contrast and resolution compared with colored images taken from new movies. Colored images have more detailes and dynamic visual features. To increase system ability to hide information in different types of images; images from old and new movies were combined to allow to

evaluate the system on images have different levels of visual complexity and resolution [38]. The variety in poses, lighting conditions, obstacles, makeup, facial modifications, and age changes in these images improves the system's ability to generalize across a wide variety of real-world contexts.

Images categorized into two classes; cover images and hidden images. Then stratified sampling was applied, guaranteeing that training, validation and testing sets splits each maintain an equal proportion of both classes. In steganography, where the goal is to conceal sensetive information within cover images, stratified sampling is employed to ensure a balanced and uniform distribution of data. This approach improves the reliability of the model evaluation and enhance the model's ability to generalize. So the proposed dataset is divided into three subsets (training 50% of the data, validation 25% and testing 25%). To standarize the image dimentions prior to embedding, all images are resized to 256x256 pixels using bilinear interpolation, and the dataset is shuffled to eliminate any ordering biases.

3.2 **Design of Models:** Generative Adversarial Networks (GANs) consist of two separate networks; the discriminator, which is a classification model, and the generator that generates samples from a noise vector [39]. This kind of model is based on game theory where these two networks are pitted against each other. The generator tries to fool the discriminator while the discriminator tries to classify between fake and real samples produced by a generator [40]. To enhance the network's ability to capture multi-scale features from the cover images, several Inception modules have been incorporated into the encoder-decoder architecture. The encoder hides the hidden image in the cover image, and the decoder reconstructs the hidden image from the stego-image. The proposed architecture uses a novel design, which is one of the key contributions of this research. In this architecture the embedded performance is enhanced by incorporating the Inception modules with residual connections. This also improves the network's ability to extract complex spatial hierarchies and patterns from cover images, ensuring efficient information concealment.

3.3 Training: In this process several optimization methods are used like Adam, Adadelta, SparseAdam, Nosadam, L-BFGS, Adamp, Tadam, RAdamplus, RMSprop, Ftrl, Nadam, AdaGrad, AdamW, Adamwt, Radam, and SGD at different rates of learning (0.1, 0.01, 0.001, and 0.0001) and batch sizes (32, 64, 128, and 256). This paper carries out a systematic exploration of these combinations to optimize the performance of the model. During the training procedure, the reconstruction loss between original hidden image and stego-image is minimized. Table 1 summarizes the hyperparameter settings used during the training process:

Table 1: Hyperparameter Selection in Training phase.

Hyperparameter	Values Explored
Optimization Methods	Adam, Adadelta, SparseAdam, Nosadam, Adamp, RAdamplus, L-BFGS, RMSprop, Tadam, Ftrl, Nadam, AdamW, AdaGrad, Adamwt, Radam, SGD
Learning Rates	0.1, 0.01, 0.001, 0.0001
Batch Sizes	32, 64, 128, 256

3.4 Performance Evaluation: SSIM, MAE, PAD, MSE, and PSNR are some of the metrics used to calculate the performance of the GAN network. Also they are used to measure the precision, perceptual quality, and fidelity of stego images. For detailed formulas and metric definitions, please refer to Section 5, where these evaluation criteria are thoroughly explained.

3.5 Result Analysis: To provide a detailed insights on the strengths and weaknesses of the proposed method; an analization of the results obtained from proposed system, along with a discussion of the effectiveness of the proposed architecture and how tuning hyperparameters affects concealing information within cover images while preserving visual quality.

4. Proposed Architecture

This study proposes a GAN based architecture for image steganography using an encoder-decoder framework. The encoder embeds the hidden image into the cover image to produce a stego image while the decoder reconstructs the hidden image from stego image. To enhance feature extraction, the encoder incorporates multiple Inception modules with parallel convolutional layers of different kernel sizes, to enable effective multi scale feature learning. The overall system design aims to balance imperceptibility and accurate recovery, as shown in Figure 2 below.

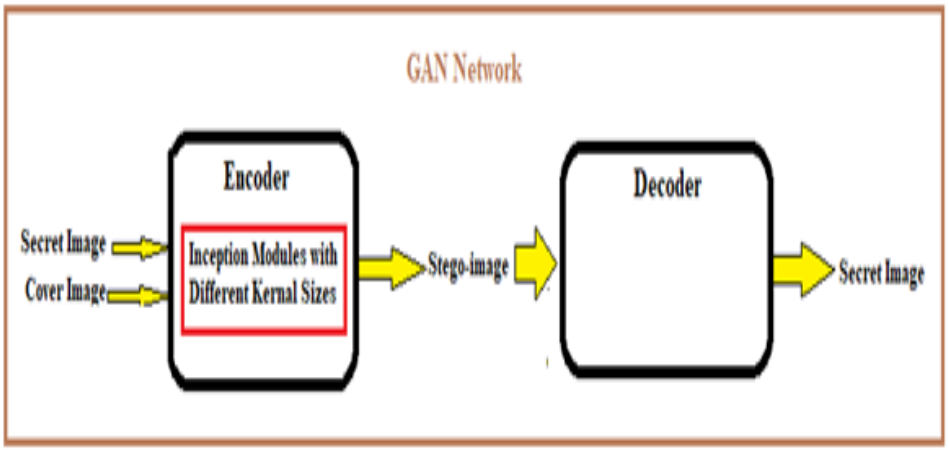

Figure 2: General System Architecture**.**

4.1 Inception Module Architecture

The Inception architecture first used in GoogLeNet. It is used to process the input using multiple filters with different kernel sizes at the same time. In this work it is applied to both the cover image and the hidden images. The goal is to extract features at different levels (multi-scale features), which means the model can capture both fine detailes and broader patterns. Then the extracted features are passed to the encoder which uses them to generate the final stego image. Table 2 presents the main components of the proposed Inception Module. Figure 3 shows the structure of the Inception Module. Figure 4 illustrates the overall system design in the form of block diagram.

Table 2: Components of Inception Module.

No.	Inception Module Components	Aim
1	1x1 Convolution	Capture local features, reduce the number of input channels (bottleneck layer)
2	3x3 Convolution	Capture medium scale features with larger receptive field
3	5x5 Convolution	Capture global features with larger receptive field
4	Residual Connection	Mitigate information loss during convolutions by maintaining channel consistency
5	ReLU Activation Function	Introduce non-linearity to the activations after each convolution
6	Summation Operation	Combine the outputs of all convolutions and the residual connection through the element wise sum

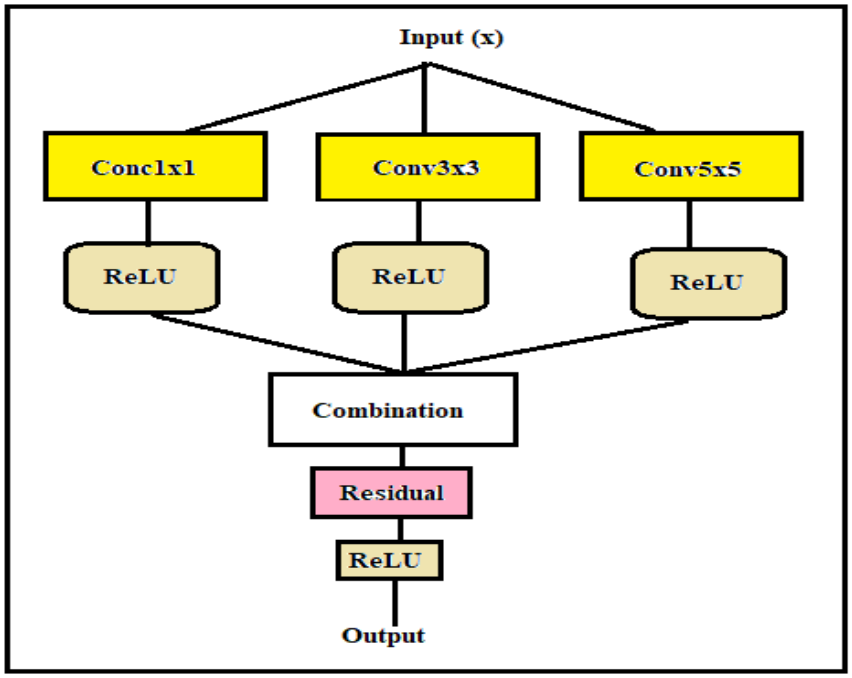

Figure 3: A Diagram of the Inception Module Architecture.

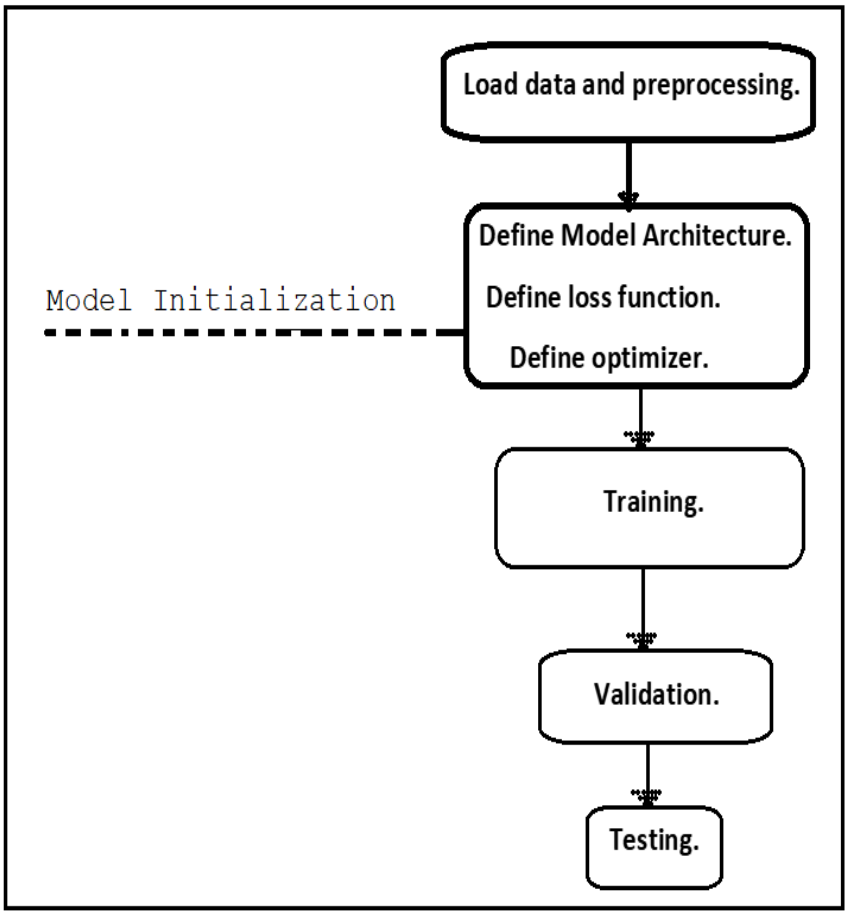

Figure 4: The Proposed System Block Diagram.

The algorithm of the proposed GAN network is shown below.

Algorithm 1: GAN System.

step 1. Data Preparation:
- Load cover and secret image datasets.
- Apply transformations (resize, to tensor).
- Creation of data loaders (train, validation, test).

step 2. Model Initialization:
- Create ISGAN model (Encoder + Decoder).
- Define loss function (MSE, Adversarial Loss).
- Choose optimizer (Adam, AdaBelief, Ftrl, etc.).

step 3. Training Loop:
FOR each epoch:
FOR each batch in train_loader:
- Get cover and secret images.
- Resize cover image if necessary.
- Forward pass: stego_image, revealed_secret = model(cover_image)
- Calculate loss (between stego_image and a randomly generated image of the same size).
- Backpropagate loss.
- Update model parameters using optimizer.

step 4. Validation Loop:
FOR each batch in val_loader:
- Get cover and secret images.
- Forward pass: stego_image, revealed_secret = model(cover_image)
- Calculate performance metrics (PSNR, SSIM, MSE, PAD, MAE).
- Compute combined metric.

step 5. Testing Loop:
FOR each batch in test_loader:
- Get cover image.
- Forward pass: stego_image, revealed_secret = model(cover_image)
- Save stego images.
- Visualize and save cover, secret, and stego image scenes.

step 6. Helper Functions:
- calculate_psnr(img1, img2): Calculate Peak Signal-to-Noise Ratio.
- calculate_ssim(img1, img2): Calculate Structural Similarity Index.
- calculate_mse(img1, img2): Calculate Mean Squared Error.
- calculate_pad(img1, img2): Calculate Pixel Absolute Difference.
- calculate_mae(img1, img2): Calculate Mean Absolute Error.
- save_image(image_np, filename): Save image as PNG.
- normalize_min_max(value, min_val, max_val): Normalize value to [0, 1].

step 7. Hyperparameter Tuning:
- Experiment with different learning rates, batch sizes, and optimizers.
- Adjust the number of Inception blocks in the Encoder.
- Tune the weights for the combined metric.

step 8. Model Saving:
- Save the trained model after training.

4.2 Encoder Architecture

Encoders can be used for different purposes inside GAN networks, ex. It can be used to generate stego images with low distortion [41], i.e., the stego images closely match the original cover images.

The encoder network converts cover images into stego images by incorporating hidden images. Using several inception modules with increasingly larger filters, the encoder extracts features from the concealed image. This is accompanied by a rise in the number of channels for each level. Finally, 1x1 convolutions are present in order to reduce the number of channels. Next, the tanh activation function is applied as given by Eq. (1) [42] on the output.

$$f(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} \tag{1}$$

Additionally, ReLU activation is applied at the output of each convolutional layer by the encoder. The ReLU function is defined using Eq. (2) [43]:

$$f(z) = \begin{cases} z, & z > 0 \\ 0, & z \leq 0 \end{cases}, or \;\; f(z) = \max(0, z) \tag{2}$$

4.3 Decoder Architecture

The decoder network comprises 2 convolutional layers and an activation function of type sigmoid. To rebuild the hidden image from the stego image, the decoder uses the sigmoid activation function placed after convolutional layers (of 3x3 and 1x1 kernal sizes).

4.4 The GAN Model

The proposed GAN model combines the encoder and decoder networks. The encoder takes a cover image and a hidden image as an input and produces a stego image that embeds the hidden data. The decoder uses the stego image to extacts the hidden image from it. To improve memory efficiency during training, a checkpoint is used in the forward pass to help reduce memory usage while computing gradients.

4.4.1 Training Loop and Hyperparameter Tuning

The the grid search technique is used to systematically test different combinations of optimizers, batch sizes, rates of learning, number of inception blocks, and other parameters. The proposed model is trained using 17 different optimization methods with 4 different learning rates values and 4 different batch size values as mentioned earlier. During the training loop several metrics are calculated to control the performance such as SSIM, PSNR, PAD, MSE, and MAE. Finally all these metrics are combined into a single combined performance metric as shown in Eq. (3) below:

$$\text{Combined metric} = \text{alpha} * \text{PSNR} + \text{beta} * \text{SSIM} \tag{3}$$

Where alpha and beta control how much weight is given to these two metrics.

Then another combined metric is used after normalizing these values to enhance the results as shown in Eq. (4) and Table 3 below:

$$\begin{aligned} \text{Combined metric} = w1 * \text{Normalized PSNR} + w2 * \text{Normalized SSIM} + \\ w3 * \text{Normalized MSE} + w4 * \text{Normalized PAD} + \\ w5 * \text{Normalized MAE} \end{aligned} \tag{4}$$

Where:

- w1,w2,w3,w4,w5 are the weights assigned to each of the normalized metrics (PSNR, SSIM, MSE, PAD, MAE), based on their relative importance to the research objectives.
- The metrics (PSNR, SSIM, MSE, PAD, MAE) are normalized as shown in Eq. (5) to Eq. (9) below to ensure that all metrics are on the same scale before being combined. This is necessary because each metric has a different range of values.

$$\text{Normalized (PSNR)} = (\text{PSNR} - \min(\text{PSNR})) / (\max(\text{PSNR}) - \min(\text{PSNR})) \quad (5)$$

$$\text{Normalized (SSIM)} = (\text{SSIM} - \min(\text{SSIM})) / (\max(\text{SSIM}) - \min(\text{SSIM})) \quad (6)$$

$$\text{Normalized (MSE)} = (\text{MSE} - \min(\text{MSE})) / (\max(\text{MSE}) - \min(\text{MSE})) \quad (7)$$

$$\text{Normalized (PAD)} = (\text{PAD} - \min(\text{PAD})) / (\max(\text{PAD}) - \min(\text{PAD})) \quad (8)$$

$$\text{Normalized (MAE)} = (\text{MAE} - \min(\text{MAE})) / (\max(\text{MAE}) - \min(\text{MAE})) \quad (9)$$

Benefits and Advantages:

1. **In-Depth Evaluation:** The suggested combined metric which is an integration of multiple metrics such as PSNR, SSIM, MSE, PAD, MAE guarantees a complete evaluation of the model effectiveness and performance, capturing both structural similarity and error-based measures simultaneously.

2. **Fairness Through Normalization:** The normalization procedure standardizes all the metrics by bringing them to the same scale, and by preventing any one metric from dominating the evaluation due to different value ranges. This allows for a more balanced and unbiased performance assessment.

3. **Flexibility and Customization:** The application of weighted factors (w1, w2, w3, w4, w5) offers researchers the flexibility to prioritize certain evaluation aspects over others based on the research goals. This adaptability enhances the robustness and applicability of the proposed approach across different problem domains.

4. **Enhanced Model Optimization:** By considering both higher-is-better (PSNR, SSIM) and lower-is-better (MSE, PAD, MAE) metrics in a unified framework, the combined metric encourages models to perform well across diverse evaluation criteria, leading to more reliable and robust solutions.

5. Performance Evaluation

A range of metrics is used by comparing the fidelity and perceived quality of the stego-images to the cover images to evaluate the steganographic performance:

1-Peak Signal-to-Noise Ratio (PSNR): Is used to evaluate and compare visual quality amongst the cover image and the stego image and is shown in Eq. (10) [44] below:

$$\text{PSNR} = 10 * \log_{10}(\text{MAX}^2 / \text{MSE}) \quad (10)$$

Where: MAX is the maximum possible pixel value (for 8-bit images; it is 255). And MSE is the Mean Squared Error between the cover image and the stego image or hidden image. If PSNR is high, then images are more similar to each other.

2-**Structural Similarity Index Measure (SSIM):** This metric is sued to compare how similar two images are between cover image and stego image. It is used to get similarity and visual quality by examining the brightness of the image (luminance), the difference in intensity between pixels (contrast), and the arrangement and patterns of pixels in the image (structural differences). Eq. (11) [45] of SSIM is shown below:

$$\text{SSIM}(x,y) = \frac{(2\mu_x \mu_y + C1)(2\sigma_{xy} + C2)}{(\mu_x^2 + \mu_y^2 + c1)(\sigma_x^2 + \sigma_y^2 + c2)} \quad (11)$$

Where:
$\boldsymbol{\mu_x}$ is the mean of the pixel values in image *x*, and $\boldsymbol{\mu}_y$ is the mean of pixels in image *y*.
$\boldsymbol{\sigma_x^2}$ and $\boldsymbol{\sigma_y^2}$ represent the variance of *x* and *y* which measure how much the pixel values vary from their mean.
$\boldsymbol{\sigma_{xy}}$ denotes the covariance between *x* and *y*; indecating how the two images change together.
$C_1=(k_1 L)^2$, $C_2=(k_2 L)^2$, are small constants added to prevent division by very small numbers, which can destabilize the calculation.
L is the dynamic range of the pixel values(e.g. 0-255 for 8 bit images).
$K_1 = 0.01$ and $k2 = 0.03$ defaulted.

3-Mean Squared Error (MSE): This measure used to quantify how different stego image is from its cover image and is shown in Eq. (12) [46] below:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(I_i - K_i)^2 \quad (12)$$

Where: N is the total number of pixels. K_i is the intensity of the i-th pixel in the cover image or hidden image. I_i is the intensity of the i-th pixel in the stego image. A smaller MSE indicates that the stego image is more similar to to the cover image, meaning the hidden data has less visibly altered the original image.

4- **Mean Absolute Error (MAE):** Compute the average discrepancy between the cover image and the stego image. A lower MAE value suggests higher similarity between images, suggesting minimal changes introduced by the steganographic embedding process. The MAE formula for Eq. (13) or less [47]:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|C_i - S_i| \quad (13)$$

Where N, S and C represent pixels total Number, Stego image, and Cover image.

5-Peak Absolute Difference (PAD): The maximum absolute difference between relevant pixels in the cover and stego images is determined by this metric as shown in Eq. (14) below [48]:

$$PAD = max_{i,j}\,|\text{C}(i,j) - \text{S}(i,j)| \quad (14)$$

Where pixel(i,j) is the cover image intensity and is denoted by C(*i,j*) while pixel(i,j) is the stego image intensity and is denoted by S(i,*j*). When there are noticeable changes between stego and cover images; I.e. when pixels in the stego image are significantly different from the corresponding pixels in the cover image this will be reflected on high PAD value and this will decrease the performance of the steganography system.

5. Results and Discussion

A comparison between the original combined metric (based on PSNR and SSIM) and the new combined metric is provided in Table 3, which normalizes multiple evaluation metrics (PSNR, SSIM, MSE, PAD, MAE) as shown in Eq. (4). Below are key observations and benefits of using the normalized combined metric:

Table 3: Hyperparameter Tuning Results from the Proposed Architecture Using Different Optimization Algorithms.

No.	Optimizer	Old PSNR dB	Old SSIM	Old MSE	Old PAD	Old MAE	Old Combined Metric	New PSNR dB	New SSIM	New MSE	New PAD	New MAE	New Combined Metric
1	Adadelta	62.18	0.5726	0.0414	0.1749	0.1749	**31.38**	59.46	0.5173	0.0824	0.2240	0.2240	**34.61**
2	Adam	62.05	0.5799	0.0439	0.1719	0.1719	**31.32**	58.61	0.4369	0.0996	0.2512	0.2512	**33.23**
3	Nosadam	61.97	0.5786	0.0448	0.1727	0.1727	**31.27**	52.88	-0.6066	0.3470	0.5544	0.5544	**16.38**
4	SparseAdam	61.94	0.5786	0.0454	0.1731	0.1731	**31.26**	58.22	0.3898	0.1083	0.2652	0.2652	**32.45**
5	Adamp	60.52	0.5329	0.0677	0.2015	0.2015	**30.52**	56.40	-0.1550	0.1607	0.3471	0.3471	**23.89**
6	L-BFGS	58.44	0.3571	0.1085	0.2637	0.2637	**29.40**	53.04	-0.6023	0.3349	0.5430	0.5430	**16.48**
7	RAdamplus	58.28	0.3310	0.1119	0.2691	0.2691	**29.30**	53.31	-0.5903	0.3164	0.5248	0.5248	**16.72**
8	Tadam	58.27	0.3288	0.1123	0.2699	0.2699	**29.30**	57.92	0.3439	0.1155	0.2768	0.2768	**31.69**
9	RMSprop	57.32	0.0857	0.1372	0.3103	0.3103	**28.70**	52.72	-0.6050	0.3613	0.5648	0.5648	**16.37**
10	Nadam	57.12	0.0079	0.1430	0.3197	0.3197	**28.56**	62.27	0.6192	0.0411	0.1680	0.1680	**36.71**
11	Ftrl	55.90	-0.3474	0.1846	0.3793	0.3793	**27.77**	51.50	-0.7500	0.4100	0.6000	0.6000	**15.90**
12	AdaGrad	54.60	-0.5012	0.2435	0.4500	0.4500	**27.05**	57.26	0.1869	0.1333	0.3052	0.3052	**29.20**
13	AdamW	54.37	-0.5172	0.2559	0.4639	0.4639	**26.93**	59.15	0.4949	0.0882	0.2330	0.2330	**34.21**
14	Adamwt	54.27	-0.5219	0.2608	0.4688	0.4688	**26.87**	57.34	0.2057	0.1314	0.3018	0.3018	**29.50**
15	SGD	54.11	-0.5311	0.2708	0.4790	0.4790	**26.79**	59.55	0.5238	0.0808	0.2217	0.2217	**34.73**
16	RAdam	53.97	-0.5375	0.2785	0.4880	0.4880	**26.72**	55.36	-0.4011	0.2006	0.4017	0.4017	**19.99**
17	Naturalgrad	52.79	-0.5727	0.3599	0.5646	0.5646	**26.11**	56.64	-0.0598	0.1523	0.3350	0.3350	**25.37**

1. Improved Differentiation of Optimizers

- In steganography the trade-off between the robstness and the imperceptibilty is affected by the choice of the optimizer. As seen, the old combined metric relying on PSNR and SSIM led the optimizers to have close scores and this makes it difficult to distinguish their effectiveness.
- Example: Under the old metric, Adadelta, Adam, and Nosadam had similar scores (31.38, 31.32, 31.27), implying comparable performance. However, the new metric reveals that Nosadam performs significantly worse (16.38), highlighting its weaknesses in error-based metrics. This means that despite Nosadam had high PSNR but its steganographic robustness is low.

2. Better Representation of General Performance

- Some optimizers performed well in PSNR but poorly in other metrics. The original combined metric did not fully capture these trade-offs.
- Example:
  - **RMSprop (Old Score: 28.70 → New Score: 16.37)**
  - RMSprop had a relatively high PSNR (57.32 dB), but its SSIM and error metrics were poor. The new combined metric lowered its ranking and making it less effective overall.

- Conversely, some optimizers that had moderate PSNR but balanced errors gained a higher score.
  - **SGD (Old Score: 26.79 → New Score: 34.73)**
  - Although SGD had a lower PSNR before, its well-balanced error metrics improved its overall ranking.

3. More Robust to Outliers in PSNR or SSIM

- In the original metric, an optimizer with high PSNR but extremely low SSIM could still have a misleadingly good score.
- The new metric prevents this by normalizing all values and considering **error-based metrics (MSE, PAD, MAE)**, ensuring that a model with poor generalization is penalized.
- Example:
  - **Ftrl (Old Score: 27.77 → New Score: 15.90)**
  - Ftrl had a high PSNR but the lowest SSIM (-0.7500) and very high error metrics. The new metric reduces its ranking accordingly.

4. Highlights Best Performers More Clearly

- The **top-performing optimizers** in the new metric are more distinct, showing that models with better balance across all metrics are preferred.
- **Top 3 Optimizers Based on New Combined Metric:**

1. **Nadam** (New Score: **36.71**) – Highest PSNR and well-balanced errors.
2. **Adadelta** (New Score: **34.61**) – Strong PSNR and moderate error metrics.
3. **SGD** (New Score: **34.73**) – Improved performance across metrics.

These rankings are more informative than the original, where optimizers like Adam and Adadelta appeared nearly identical.

Since PSNR nad SSIM tell how the image looks but they don't tell how well the hidden data survives or how much information can be hidden. Thus the new combined metric gives fuller evaluation of the The revised rankings highlight optimizers that ensure a better balance between image quality and hidden data preservation, guiding the selection of optimal algorithms for steganographic applications. Figure 5 studies the effect of batch size on image quality and imperceptibility, it shows a comparison of stego images using high rate of learning = 0.1 with different batch sizes (32, 64, 128). Figure 6 focuses on the effect of the optimizer, it shows a comparison using a very low learning rate of 0.0001 and large batch size 256 with different optimizers, highlighting the impact of optimization methods on the performance.

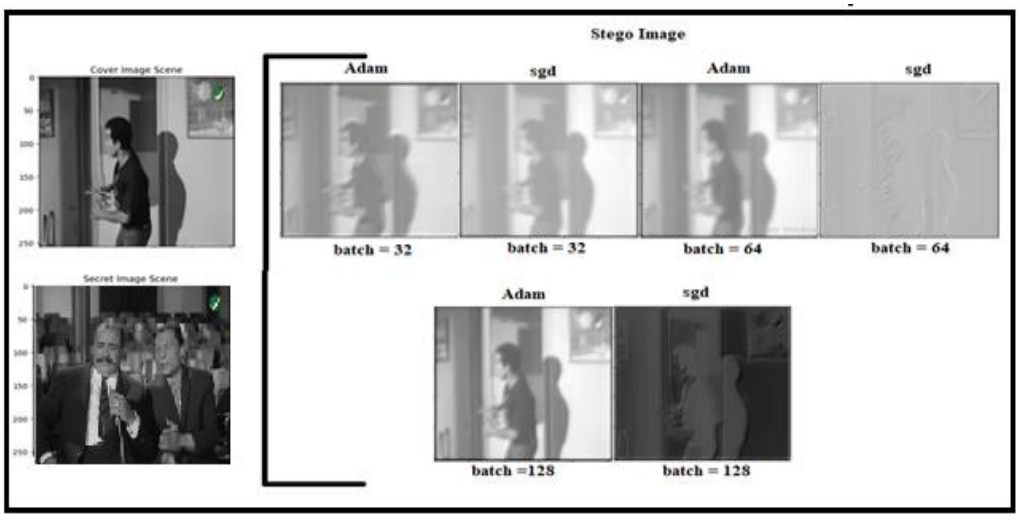

Figure 5: A Comparative Analysis of Stego Images Using Batch Sizes (32, 64, and 128) and LR=0.1 with SGD and Adam Optimization Methods.

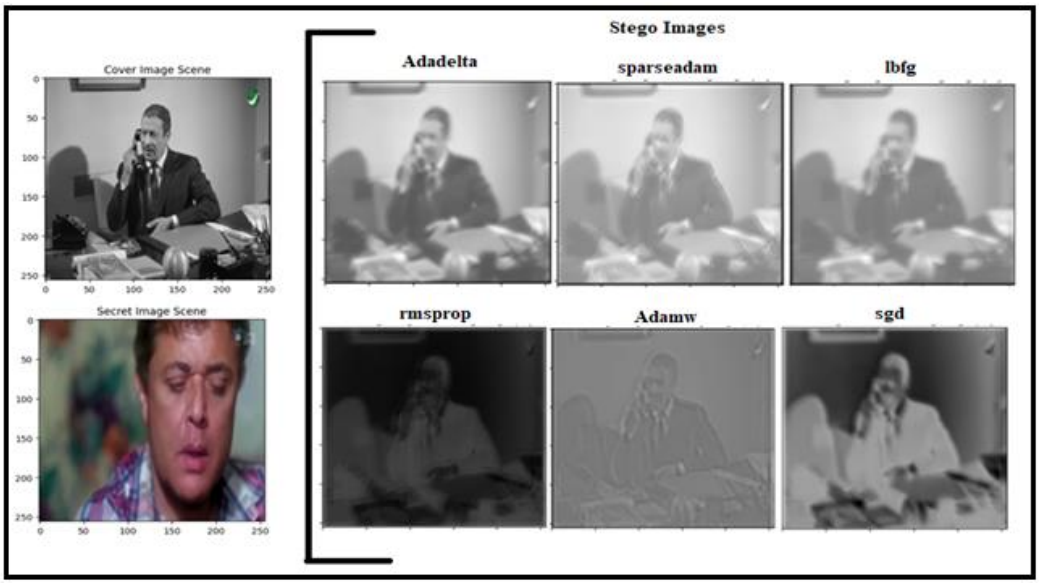

Figure 6: A Comparative Analysis of Stego Images Using Batch Sizes =256 and LR= 0.0001 with Different Optimization Methods.

6. Conclusions and Future Work

A novel steganography system is presented with Generative Adversarial Network (GAN) architecture and hyperparameters optimization. This work focuses on enhancing imperceptibility which is a crucial aspect of steganographic security. The proposed system achieves robust hidden image embedding with minimal distortion to the cover image, facilitated by the use of multi-scale feature extraction using Inception modules in the encoder. This approach significantly enhances the system's resilience against steganalysis. The improved imperceptibility raised the values of PSNR and SSIM, makes it considerably more challenging for steganalysis methods to detect the hidden information.

Furthermore, this work introduces a comprehensive evaluation of steganographic performance by using a new combined metric (incorporating MSE, PAD, MAE, PSNR and SSIM). This metric addresses the limitations of relying solely on PSNR and SSIM, which can be misleading as they don't fully capture the balance between imperceptibility and robustness. The obtained results from Table 3 showed that using Nadam and Adadelta as optimizers led to strong results and produced stego images that looked visually excellent. Therefore in this GAN application; these optimizers are considered to be the best. Conversely, the new combined metric highlighted Ftrl and RMSprop limitations for steganography because they had high PSNR but poor performance in other metrics. The use of this improved evaluation metric allows for a more nuanced understanding of optimizer performance and facilitates the selection of algorithms that truly optimize for robustness and imperceptibility.

On the other hand, the proposed evaluation metric showed that optimization algorithms like Naturalgrad, RAdam, and SGD, had overall limited effectiveness because they exhibited low PSNR values and negative SSIM scores. The use of optimizers identified by the new metric as effective enhanced the level of imperceptibility proposed system which considered a significant challenge to conventional steganalysis methods that depend on detecting statistical irregularities caused by using embedded data. Also the encoder improved the security by making the hidden data less detectable and more resistant to attacks because its architecture included tanh activation function and 1x1 convolutions. Tanh activation function adds non-linearity so making the relashionship between the input and the output more complex and harder to be predicted. While, 1x1 convolutions reduced the dimentionality of the hidden message by mixing its features.

The diversity in the dataset (such as different poses, lighting, obstacles, and age) helped the system learn to be more robust, better at hiding data (I.e. concealment without detecting) and work under different real world conditions.

Future work can be done on several key areas:

- Exploring further refinements of the Inception modules: Investigating different Inception module configurations to optimize feature extraction for steganography.
- Additional advanced optimization algorithms can be evaluated and integrated: The proposed combined metric can be used to test and compare other different optimization algorithms to select the ones that make the proposed system with best performance.
- Investigating adaptive embedding strategies: Depending on images features; choose the best regions in an image for hiding secret information.
- Extending the system by handling different data types: The system can be developed to hide data not only in images but also in text files, audio files or videos, while maintaining high levels of imperceptibility.

References

[1] A. M. Alejandro, H. Alfonso, A. Moutaz, J. Jason, et al. "Evolving Generative Adversarial Networks to Improve Image Steganography". *Expert Systems with Applications*, vol. 222, pp. 119841-119855, Jul. 2023, doi: 10.1016/j.eswa.2023.119841.

[2] K. Muna and N. D. Ban, "Deep Learning Based on Attention in Semantic Segmentation: An Introductory Survey," *Iraqi Journal of Computers, Communications, Control & Systems Engineering (IJCCCE)*, vol. 23, no. 1, pp. 104-114, Mar. 2023, doi: 10.33103/uot.ijccce.23.1.9.

[3] M. Daudi, K. Herald, K. Ellen, and H. Ndyetabura,"Deep learning approaches for fault detection and classifications in the electrical secondary distribution network: Methods comparison and recurrent neural network accuracy comparison," *Cogent Engineering*, vol. 7, no. 1, pp. 1-25, 2020, doi: 10.1080/23311916.2020.1857500

[4] T. Satvik, I. A. Alisha, D. Adam, S. Rithvik, et al., "Recent advances and application of generative adversarial networks in drug discovery, development, and targeting," *Artificial Intelligence in Life Sciences*, vol. 2, no. 6, pp. 100045-100066, 2022, doi: 10.1016/j.ailsci.2022.100045

[5] N. Mohammed, A. Mohammed, and A. Tanvir, "Prediction of concrete compressive strength using deep neural networks based on hyper parameter optimization," *Cogent Engineering*, vol. 11, no. 1, pp. 1-14, Feb. 2024, doi: 10.1080/23311916.2023.2297491.

[6] M. V. Crallet, N. M. Alloys, and I. M. Salehe, "A review on deep learning aided pilot decontamination in massive MIMO," *Cogent Engineering*, vol. 11, no. 1, pp. 1- 13, Feb. 2024, doi: 10.1080/23311916.2024.2322822.

[7] I. A. Asmah and P. M. Erwin, "Classification of Diseases in Oil Palm Leaves Using the GoogLeNet Model," *Baghdad Science Journal*, vol 20, no. 6 Suppl., pp. 2508-2520, 2023, doi: 10.21123/bsj.2023.8547.

[8] T. Christian, "Catch Me If You GAN: Using Artificial Intelligence for Fake Log Generation," arXiv, vol. 2112.12006, pp. 1-6, Dec. 2021, doi: 10.48550/arXiv.2112.12006.

[9] K. Aechan, P. Mohyun, and H. L. Dong, "Ai-ids: Application of deep learning to real time web intrusion detection," *IEEE Access*, vol. PP, no. 99, pp. 1–1, 2020, doi: 10.1109/ACCESS.2020.2986882.

[10] G. Gourav, R. Kiran, G. Manish, J. Tony, and et al., "A Comprehensive Review of Deep Fake Detection Using Advanced Machine Learning and Fusion Methods," *Electronics*, vol. 13, no. 1, pp. 95-122, 2024, doi: 10.3390/electronics13010095.

[11] P. Sayantica and A. L. Simone, "Encryption Based on Neural Cryptography," in *Hybrid Intelligent Systems*, pp. 321–33, 2018, doi: 10.1007/978-3-319-76351-4_33.

[12] W. M. Arif and S. Bisma, "Deep learning based image steganography: A review," *Wiley Interdisciplinary Reviews: DataMining and Knowledge Discovery*, vol. 13, no. 3, pp.1481-1507, 2023, doi: 0.1002/widm.1481.

[13] J. Khandelwal and V. K. Sharma, "Reversible Image Steganography Using Deep Learning Method: A Review," in *Proceedings of the International Conference on Human-Centric Smart*

Computing (ICHCSC 2023), vol. 376, *Smart Innovation, Systems and Technologies (SIST)*, pp. 625-635, 2024, doi: 10.1007/978-981-99-7711-6_49.

[14] K. Hamza, H. Mustapha, H. Yassine, M. David, A. Abbes, "Deep Learning for Diverse Data Types Steganalysis: A Review of methods, taxonomy, challenges and future directions," *Neurocomputing*, vol. 581,no. 3, pp. 127528- 127567, May 2024, doi: 10.1016/j.neucom.2024.127528.

[15] S. Anurag, K. Ankur , R. Deepak, J. Subhanshu, et al., "Survey on Encoding Binary Data within a Digital Image Using Deep Steganography and Multilayered Neural Network," *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, vol. 11, no. 3, pp.1515- 1519, Mar. 2023, doi: 10.22214/ijraset.2023.49742.

[16] Y. A. Sharing, E. T. Nada, Q. A. Mohammed, "An Enhanced Approach of Image Stenographic Using Discrete Shearlet Transform and Secret Sharing," *Baghdad Science Journal*, vol. 19, no. 1, pp. 197-207, 2022, doi: 10.21123/bsj.2022.19.1.0197.

[17] H. J. Mithal, A. J. Fanar, A. N. Mohammed, "Building a Statistical Model to Detect Foreground Objects and using it in Video Steganography," *Baghdad Science Journal*, vol. 20, no. 6, 2023, pp. 2330-2341, doi: 10.21123/bsj.2023.ID.

[18] A. Fatimah, U. Gary, M. Graham, "Improving Detection of Deep Fakes through Facial Region Analysis in Images," *Electronics*, vol. 13, no. 1, pp. 126-148, Dec.2023, doi: 10.3390/electronics13010126.

[19] Ş. Yusuf, K. Mahmut, D. Recep, "Channel Hopping Steganography for Color Images," in *Proceedings of the 7th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pp. 1-4, Nov. 2023, doi: 10.1109/ISMSIT58785.2023.10304914.

[20] H. Apichat, K. Yossawee, "Four enhanced algorithms for full size image hiding in chest x-ray images," *Multimedia Tools and Applications*, 2024, doi: 10.1007/s11042-024-18226-8.

[21] S. Ahmed and A. Hussein, "High Security and Robustness Image Steganography Based on Ant Colony Optimization Algorithms and Discrete Cosine Transform," *Journal College of Education for Pure Science (JCEPS)*, vol. 13, no. 4, pp. 118- 150, Dec. 2023, doi: 10.32792/utq.jceps.10.01.01.

[22] B. Mariusz and K. Grzegorz, "Hiding Information in Digital Images Using Ant Algorithms," *Entropy*, vol. 25, no. 7, pp. 963-993, Jun. 2023, doi: 10.3390/e25070963.

[23] S. Eman, S. Hind, A. Inas, "Advanced Intelligent Data Hiding Using Video Stego and Convolutional Neural Networks,". *Baghdad Science Journal*, vol. 18, no. 4, pp. 1317-1327, April 2021, doi: 10.21123/bsj.2021.18.4.1317.

[24] M. Nachaat, "Current trends in AI and ML for cyber security: A state-of-the-art survey," *Cogent Engineering*, vol. 10, no. 2, pp. 1-30, Sep. 2023, doi: 10.1080/23311916.2023.2272358.

[25] M. Faizan, A. Shoaib, I. Talha, G. Christer, and A. Hazarat, "Segmentation of lungs in chest X-Ray image using generative adversarial networks" *IEEE Access*, vol. 8, pp. 153535–153545, Aug. 2020, doi: 10.1109/ACCESS.2020.3017915.

[26] C. Zhiwu, W. Wenjing, E. Qing, L. Yingbo, et al., "A Denoising Method for Multi-Noise on Steel Surface Detection", *Applied Sciences*, vol. 13, no. 18, pp. 10471-10488, 2023, doi: 10.3390/app131810471.

[27] M. Francesco, B. Luca, M. Fabio, S. Antonella, "Generative Adversarial Networks in Retinal Image Classification," *Applied Sciences*, vol. 13, no. 18, pp. 10433-10452, Sep. 2023, doi: 10.3390/app131810433.

[28] G. Ian, P. Jean, M. Mehdi, X. Bing, et al., "Generative Adversarial Networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139-144, Oct. 2020, doi: 10.1145/3422622.

[29] D. N. Sindhura , M. P. Radhika, N. B. Shyamasunder, P. M. Manohara, "Deep learning-based automated spine fracture type identification with Clinically validated GAN generated CT images," *Cogent Engineering*, vol. 11, no. 1, pp. 1-12, Jan. 2024, doi: 10.1080/23311916.2023.2295645.

[30] T. W. Rima, Y. Indah , A. S. Indah, D. S. Budi, and et al., "Improvement of chest X-ray image segmentation accuracy based on FCA-Net," *Cogent Engineering*, vol. 10, no. 1, pp. 1-16, 25 June 2023, doi: 10.1080/23311916.2023.2229571.

[31] M. Maha and Y. Mohammed, "Diagnosis of COVID-19 from X-ray images using deep learning techniques," *Cogent Engineering*, vol. 9, no. 1, pp. 1-27, Sep. 2022, doi: 10.1080/23311916.2022.2124635.

[32] M. Loey, F. Smarandache, N. Khalifa, "Within the lack of chest COVID-19 X-ray dataset: A novel detection model based on GAN and deep transfer learning, "*Symmetry*, vol. 12, no. 4, pp. 651-670, Apr. 2020, doi: 10.3390/sym12040651.

[33] M. Momina , N. Mariam, M. Khalid, J. Ali, and et al., "Deep fakes generation and detection: State-of-theart, open challenges, countermeasures, and way forward," *Applied Intelligence*, vol. 53, pp.3974-4026, 2023, doi: 10.1007/s10489-022-03766-z.

[34] V. Denis, N. Ivan, B. Evgeny, "Stenographic generative adversarial networks," in *Twelfth international conference on machine vision (ICMV)*, vol. 11433, pp. 114333M , Jan. 2020, doi: 10.1117/12.2559429.

[35] Q. Jiaohua, W. Jing, T. Yun, H. Huajun, and et al., "Coverless image steganography based on generative adversarial network, " *Mathematics*, vol. 8, no. 9,pp. 1394-1405, Aug. 2020, doi: 10.3390/math8091394.

[36] Y. Ye, W. Junyu, C. Qi, R. Yizhi, M. Weizhi, "High invisibility image steganography with wavelet transform and generative adversarial network," *Expert Systems with Applications*, vol. 249, pp. 123540, 2024, doi: 10.1016/j.eswa.2024.123540.

[37] W. Dan, L. Ming, Z. Yushu, "Adversarial Data Hiding in Digital Images," *Entropy*, vol. 24, no. 6, pp.749-767, 2022, doi: 10.3390/e24060749.

[38] P. C. Joseph, M. Paul, D. Lan, "COVID-19 Image Data Collection," *ARXIV*, pp. 1-4, Mar. 2020, doi: 10.48550/arXiv.2003.11597. [Online]. Available: https://doi.org/10.48550/arXiv.2003.11597.

[39] Z. Huanhuan, and Q. Yufei, "Applying Deep Learning to Medical Imaging: A Review," *Applied Sciences*, vol. *13, no.* 18, pp. 10521-10546, Sep. 2023, doi: 10.3390/app131810521.

[40] A. Nuha, S. Arcot, M. Nadine, M. Gelareh, "Video Generative Adversarial Networks: A Review," *ACM Computing Surveys*, vol. 55, no. 2, Article 30, pp. 1-25, Jan. 2022, doi: 10.1145/3487891.

[41] F. Zhangjie, W. Fan, C. Xu, "The secure steganography for hiding images via GAN," *EURASIP Journal on Image and Video Processing*, vol. 110, no. 1, pp. 1-19, 2020, doi: 10.1186/s13640-020-00534-2.

[42] K. Pavel, and M. Sebastien, "Deep fake detection: humans vs. machines," *arXiv preprint*, arXiv:2009.03155, pp. 1-7, Sep. 2020, doi: 10.48550/arXiv.2009.03155.

[43] M. Asraa , and R. Noor, " Predicting Movie Production Years through Facial Recognition of Actors with Machine Learning," *Baghdad Science Journal*, vol. 22, no. 1, pp. 1-19, 2024, doi: 10.21123/bsj.2024.8996.

[44] M. Noor and E. A. Matheel, "mRNA Approach Image Encryption Using Algorithm," Iraqi Journal of Science, vol. 64, no. 5, pp. 2545-2560, 2023. doi: 10.24996/ijs.2023.64.5.37.

[45] M. Yisroel, and L. Wenke , "The Creation and Detection of Deep fakes: A Survey," *ACM Computing Surveys*, vol. 54, no. 1, Article 7, pp. 1–41, 2021, doi: 10.1145/3425780.

[46] A. Taif, and M. Wasfy, "An adaptive steganography insertion technique based on wavelet transforms," *Journal of Engineering and Applied Sciences*, vol. 70, pp. 144-172, 2023, doi: 10.1186/s44147-023-00300-x.

[47] T. Mustafa, Ö. Adem, A. Naim, T. Faruk, "Highly Secured Hybrid Image Steganography with an Improved Key Generation and Exchange for One-Time-Pad Encryption Method," *Afyon Kocatepe University Journal of Science and Engineering*, vol. 23, no. 1, pp. 101-114, 2023, doi: 10.35414/akufemubid.1128075.

[48] P. Yi-Lun, and W. Ja-Ling, "Rate-Distortion-Based Stego: A Large-Capacity Secure Steganography Scheme for Hiding Digital Images," *Entropy*, vol. 24, no. 7, pp. 982-1010, Jul. 2022, doi: 10.3390/e24070982.