## THIRD YEAR B. TECH SIXTH SEMESTER COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

Machine Learning

# Project Report

Deep Steganography



**Avinash Wagh BT21CSE108 Chirag Salwan BT21CSE110 Parth Lanjewar BT21CSE129 Prajwal Patil BT21CSE133**

**INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, NAGPUR**

**(An Institution of National Importance by Act of Parliament)**

**VisualVault : End to End encryption and decryption deep image steganography system using CNN**

Chirag Salwan Parth Lanjewar Avinash Wagh

Computer Science & Engg Department Computer Science & Engg Department Computer Science & Engg Department

IIIT Nagpur IIIT Nagpur IIIT Nagpur

Nagpur, India Nagpur, India Nagpur,India

[bt21cse110@gmail.com](mailto:bt21cse110@gmail.com) [bt21cse129@iiitn.ac.in](mailto:bt21cse129@iiitn.ac.in) bt21cse108@iiitn.ac.in

Prajwal Patil

Computer Science & Engg Department

IIIT Nagpur

Nagpur, India

[bt21cse133@iiitn.ac.in](mailto:bt21cse133@iiitn.ac.in)

**ABSTRACT:**

In today's modern world, ensuring secure transmission and reception of data is of paramount importance yet remains a challenging task. While various encryption standards have been established, many rely on key-based algorithms for encryption and decryption. However, the rapid advancement of technology poses a threat to these standards, rendering them vulnerable to decryption by high-performance computing. Steganography, the art of concealing information, offers an alternative approach that has evolved over the years. One prevalent method in contemporary image steganography involves utilizing the least significant bit (LSB) to embed data within an image. Nevertheless, these techniques are susceptible to detection and decryption using existing technologies.

This paper introduces an end-to-end encoder-decoder framework akin to deep image steganography systems. It conceals an image within another image, employing methods resilient to contemporary attacks. Furthermore, the resultant image undergoes enhancement through processing via the SRGAN/ESRGAN system, aiming for superior image resolution. Additionally, users have the option to generate the cover image using stable diffusion techniques, enhancing the overall security and robustness of the system. By combining advanced steganographic methods with state-of-the-art image enhancement techniques, this system offers a robust solution for secure data transmission in the face of evolving technological landscapes.

**Key words :** Deep image steganography , CNN, ESRGAN, Stable Diffusion

# Introduction

## Problem Statement

Steganography is the art and science of hiding secret information within other information, such as text, images, audio, or video. Steganography has many applications in cryptography, security, and privacy, especially in the era of digital communication and media. However, traditional steganography methods often rely on simple techniques that embed data in less critical parts of the media, such as the least significant bits of pixels or samples. These methods are vulnerable to detection and distortion by various attacks, such as compression, noise, filtering, or cropping. Moreover, these methods have limited capacity and quality, as they can only hide a small amount of data without affecting the perceptibility of the media.

## Objective

The objective of this project is to develop a novel technique for steganography using deep learning algorithms, called deep steganography. Deep steganography aims to overcome the limitations and challenges of traditional steganography methods by leveraging the complexity and diversity of digital media files. Deep steganography uses deep neural networks to analyze the carrier file’s features and statistical patterns, and embed the secret data in a way that is more secure and imperceptible. Deep steganography also uses deep neural networks to extract the hidden data from the carrier file with high accuracy and fidelity. Deep steganography can achieve higher capacity and quality than traditional methods, as it can hide more data without compromising the visual or auditory quality of the media.

## Application

Deep steganography has many potential applications in various domains and scenarios, such as:

* + 1. **Secure communication:**

Secretly exchanging sensitive information in publicly accessible media, like sharing messages within images on social media. This can protect the privacy and confidentiality of the communication from eavesdroppers, hackers, or censors.

* + 1. **Digital watermarking:**

Embedding copyright information or authentication markers within media files imperceptibly. This can prevent unauthorized copying, distribution, or modification of the media, and provide proof of ownership and origin.

* + 1. **Data embedding:**

Securely storing additional data within audio or video recordings, such as sensor readings, metadata, or annotations. This can enhance the functionality and utility of the media and provide contextual information and verification.

# Literature review

* 1. **HIDING IMAGES IN PLAIN SIGHT: DEEP STEGANOGRAPHY**

The paper discusses the practice of steganography, which involves concealing a secret message within another message. Specifically, the paper focuses on hiding a full-size colour image within another image of the same size using deep neural networks. The system is trained on images from the ImageNet database and works well on natural images from various sources. Unlike traditional steganographic methods that encode the secret message within the least significant bits of the carrier image, this approach compresses and distributes the secret image's representation across all available bits.

The paper describes the architecture of the system, which consists of three components: the Prep-Network [1], the Hiding Network [1], and the Reveal Network [1]. These components are trained as a single network and work together to prepare the secret image, hide it within the cover image, and reveal it to the receiver. The paper also discusses the error propagation in the system and evaluates its performance through empirical testing [1].

The results show that the system is able to hide the secret image with minimal distortion to the cover image. Additionally, the paper explores the discoverability of the hidden message and proposes methods to make it more difficult to detect. The document concludes by discussing future directions for research, including

addressing the detection of hidden messages and exploring other applications of the system, such as embedding text or audio.

The paper proposes a novel method for hiding a full-size color image within another image using deep neural networks, but it does not evaluate the robustness of the hidden images against common image processing operations, such as cropping, resizing, filtering, or compression. These operations could potentially degrade or destroy the hidden information, making it difficult or impossible to recover. Therefore, a future work could investigate how to improve the resilience of the proposed method against various image manipulations, and compare it with other steganographic techniques.

## Image Steganography: A Review of the Recent Advances

The main goal of image steganography is to make the hidden information imperceptible to human eyes. The paper discusses various deep learning methods used in image steganography, including traditional methods, convolutional neural network (CNN)-based methods, and generative adversarial network (GAN)-based methods [2]. It also provides details on the datasets commonly used for evaluation and the evaluation metrics used to measure the performance of the methods.

The document highlights the challenges in image steganography, such as the lack of benchmark datasets and the need for more research on hiding images in images and videos. It also suggests future directions for research, including exploring different network architectures, optimizing parameters, and considering security against attacks during data transfer [2].

Overall, the paper provides a comprehensive overview of recent advances in image steganography and highlights the potential of deep learning methods in this field.

The paper proposes a deep neural network approach to hide a full-size color image within another image, without significantly altering the appearance or statistics of the cover image. However, the paper does not address the robustness of the hidden image against common image processing operations, such as cropping, resizing, compression, filtering, or noise addition. These operations could potentially degrade or destroy the hidden image, making it difficult or impossible to recover. Therefore, a research gap is to evaluate the performance and limitations of the proposed method under various image processing scenarios, and to explore possible ways to enhance the resilience of the hidden image.

## Multi-Image Steganography Using Deep Neural Networks

The document discusses the implementation of multi-image steganography using deep neural networks. The goal is to hide multiple secret images within a single cover image while maintaining the quality and secrecy of the encoded messages [3]. The proposed methodology combines ideas from these papers, utilizing multiple prep and reveal networks to encode and decode the secret images.

The model architecture consists of prep networks, hiding networks, and reveal networks. The implementation details include the use of Tiny ImageNet dataset, training with Adam optimizer, and the calculation of loss for both cover and secret images [3].

The results show that the encoded cover image closely resembles the original cover image, and the secret images can be successfully retrieved with minimal loss. The document also mentions future directions for improvement, such as increasing the number of secret images, exploring different loss functions, and using conditional decoders [3]. Overall, the implementation demonstrates the potential of deep neural networks in multi-image steganography.

The paper proposes a deep neural network model for multi-image steganography, which can hide and reveal multiple secret images within a single cover image. However, the paper does not evaluate the security and robustness of the proposed method against steganalysis attacks or common image processing operations, such as compression, cropping, or filtering. Moreover, the paper does not compare the performance of the proposed method with existing steganographic methods for multi-image hiding, such as (Chen et al., 2019) or (Wang et al., 2020). Therefore, a research gap is to conduct a comprehensive security and performance analysis of the proposed method and compare it with the state-of-the-art methods in the literature.

## End-To-End Trained CNN Encoder-Decoder Networks For Image Steganography

The document discusses a novel approach to image steganography using end-to-end trained CNN encoder-decoder networks[4]. The existing methods of image steganography use manually crafted features to hide binary payloads into cover images, resulting in limited payload capacity and image distortion. The proposed approach uses a deep learning based generic encoder-decoder architecture to embed images as payload. The major contributions of this approach are: (i) the introduction of a deep learning based generic encoder-decoder architecture for image steganography[4], (ii) a new loss function that ensures joint end-to-end training of encoder-decoder networks[4], and (iii) extensive empirical evaluation on various datasets, reporting state-of-the-art payload capacity at high PSNR and SSIM values. approx. achieve this payload of 33% (on average 8 bpp) with the average PSNR values of 32.9 db. (SSIM =0.96) for cover and 36.6 db. (SSIM=0.96) for recovered payload image – c.f[4].

The encoder network takes a host cover image and a guest payload image as input and produces a hybrid output image. The goal of the encoder network is to produce a hybrid image that visually resembles the host image but also contains the content of the guest image[4]. The decoder network takes the encoder produced hybrid image as input and recovers the guest image from it. The loss function for the encoder and decoder network is a combination of the difference between the input and output images and the regularization terms for the encoder and decoder weights.

The experiments conducted on datasets such as CIFAR10, MNIST, ImageNet, LFW, and PASCAL-VOC12 demonstrate the effectiveness of the proposed algorithm. The algorithm is able to hide a significant payload in cover images while maintaining high PSNR and SSIM values. The results show that the algorithm is generic and robust, capable of handling complex backgrounds and variations in object appearance. The algorithm also exhibits generalization capabilities, as it performs well on unseen datasets [4]. Overall, the proposed approach offers a promising solution for image steganography using deep neural networks.

The paper proposes a novel CNN-based encoder-decoder architecture for image steganography, which can hide one image into another and recover it with high payload capacity and perceptual quality1. However, the paper does not evaluate the security and robustness of the proposed method against steganalysis attacks, which aim to detect the presence of hidden information in images. Moreover, the paper does not compare the proposed method with existing image steganography methods,

such as LSB, WOW, or histogram modification, in terms of embedding capacity, image quality, and steganalysis resistance. Therefore, a research gap is to conduct a comprehensive security and performance analysis of the proposed method and benchmark it with state-of-the-art image steganography methods.

### SteganoGAN: HIGH-CAPACITY image steganography with GANs

The authors start by discussing the limitations of traditional steganography techniques, which are only effective up to a certain payload and tend to introduce artifacts that can be easily detected. They then introduce the concept of deep learning-based steganography, which has shown promising results in achieving higher embedding rates. However, existing deep learning approaches have their own limitations, such as imposing constraints on the size of the cover image and not exploring the limits of how much information can be hidden successfully. To address these limitations, the paper proposes STEGANOGAN [5], a novel end-to-end modelfor image steganography that builds on recent advances in deep learning. The model uses dense connections and multiple loss functions within an adversarial training framework to optimize the encoder, decoder, and critic networks simultaneously. The authors demonstrate that their approach successfully embeds arbitrary data intocover images from various datasets and achieves state-of-the-art embedding rates of

4.4 bits per pixel while evading detection by steganalysis tools [5]. The paper also introduces new evaluation metrics for steganography algorithms, including the Reed-Solomon bits per pixel (RSBPP) metric to measure the effective number of bits that can be conveyed per pixel, and the peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) to measure image quality [5]. The authors provideexperimental results on different datasets and model variants, showing the superiority of their approach in terms of relative payload and image quality. Approx Accuracy: = 0.95, RS-BPP: = 2.85, PSNR: = 36.94, SSIM: = 0.92

In conclusion, the paper presents a comprehensive and innovative approach to image steganography using GANs, offering higher embedding rates and improved image quality compared to existing techniques. The proposed model, STEGANOGAN, demonstrates its effectiveness in hiding arbitrary data in images while evading detection by steganalysis tools.

The lack of a comprehensive evaluation of the proposed technique on different types of images and data. The paper only experiments with two datasets, Div2K and

COCO, which may not be representative of the diversity and complexity of natural images and arbitrary data. Moreover, the paper does not compare the performance of the proposed technique with other deep learning-based steganography methods, such as (Baluja, 2017) and (Wu et al., 2018), which also aim to hide images in images. Therefore, it is unclear how the proposed technique generalizes to different scenarios and how it compares with existing methods in terms of relative payload, image quality, and secrecy.

### An Optimized Steganography Hiding Capacity and Imperceptibly Using Genetic Algorithms

The proposed scheme utilizes genetic algorithms (GAs) to search for the optimum positions to hide the secret data within the carrier image. The scheme consists of several operations, including pixel scanning, pixel shifting, flipping secret bits, transposing, LSB matching checking, and secret data embedding. These operations are designed to increase the least significant bits (LSB) matching between the carrier and stego images, resulting in increased embedding capacity and reduced distortion[6]. The paper also provides a detailed explanation of genetic algorithms and their components, such as representation, evaluation of individuals, population, selection mechanism, reproduction operators, and replacement. The proposed scheme is evaluated using peak signal-to-noise ratio (PSNR) as a performance metric. Experimental results show that the proposed scheme outperforms traditional LSB schemes and other GA-based steganography schemes in terms of PSNR values[6]. The scheme is visually and statistically undetectable, and it provides high embedding capacity. The paper concludes that the proposed scheme achieves high- quality stego images while maintaining imperceptibility and security. Overall, the paper presents a comprehensive and innovative approach to steganography using genetic algorithms.

The paper only uses PSNR as a performance metric, and does not compare the proposed scheme with other metrics, such as structural similarity index (SSIM), mean squared error (MSE), or normalized correlation (NC). It is not clear how the proposed scheme performs according to these metrics, or how it can be improved to achieve better results.

### The Development and New Direction of Digital Image Steganography

The paper begins by explaining that digital image steganography is the science of hiding hidden information in a carrier image, and it has become an important choice for covert communication in various fields such as military and security. The paper then delves into the development of traditional digital image steganography, discussing techniques such as LSB steganography, multiple least significant bits (MLSB)[7], and steganography based on second-order statistical information. It also explores the use of machine learning algorithms like support vector machines for steganalysis. The paper further discusses the advancements in adaptive steganography, including techniques like HUGO and S-UNIWARD, which improve the anti-detection performance of spatial image steganography.

The paper then introduces the new directions in digital image steganography, specifically focusing on the application of deep learning and generative adversarial networks (GANs)[7]. It explains how deep learning, particularly convolutional neural networks (CNNs), can be used for steganalysis, and highlights the advantages of CNNs in learning complex representations from data. The paper also discusses the use of GANs in steganography[7], where the generator is used to generate the carrier image and then the steganography algorithm is applied to embed the secret message. It also explores the use of counter-sample correlation techniques in steganography, where carefully designed noise is added to the image to deceive deep neural networks.

In conclusion, the paper states that digital image steganography and steganalysis are developing towards more intelligent methods, with adversarial samples and adversarial networks being typical examples. It suggests that the future of digital image steganography lies in optimizing the steganography model based on counter- sample techniques and generative adversarial networks[7], in order to achieve equivalent or better performance than adaptive steganography. Overall, the paper provides a comprehensive overview of the development and new directions in digital image steganography, highlighting the advancements in techniques and the potential for future research.

### Image Steganography Analysis Based on Deep Learning

The paper begins by highlighting the challenges faced by steganalysis and the need for a more effective approach. It then introduces the use of convolutional neural networks (CNNs) for steganalysis, which involves extracting deep features from stego images[8]. The CNN model is designed to capture the numerical attributes, such as neighborhood relationships, that are important for steganalysis. The paper also explores the use of global information constraints in steganalysis, which involves applying statistical analysis to the entire image. This approach enhances the feature learning process and improves the throughput of the model. Additionally, the paper discusses steganalysis based on low embedding rate images, which are more difficult to detect. To overcome this challenge, the paper proposes a technique that leverages knowledge gained from high embedding rate images to improve detection on low embedding rate images. Finally, the paper addresses the issue of multi-class steganography, where different types of steganography algorithms are used. The paper suggests using a multi-task CNN framework to detect the various types of steganography algorithms[8]. Overall, the paper presents a comprehensive analysis of image steganography using deep learning techniques, highlighting the advantages and limitations of each approach.

Lack of comparison with other steganalysis methods: The paper proposes several steganalysis techniques based on deep learning, global information, low embedding rate, and multi-class steganography, but it does not compare their performance with other existing or state-of-the-art steganalysis methods1. It is unclear how the proposed techniques fare against the conventional or recent approaches in terms of accuracy, robustness, efficiency, and scalability. A comprehensive evaluation and analysis of the proposed techniques with other steganalysis methods would be beneficial to demonstrate their advantages and limitations.

### Spatial Image Steganography Based on Generative Adversarial Network

The research paper introduces a novel secure steganography scheme that leverages generative adversarial networks (GANs)[9]. The proposed method allows for the seamless embedding of secret information into digital images without detection by steganalysis techniques. Key components of the scheme include a Tanh-simulator function for optimal embedding simulation, a U-Net based generator to learn embedding change probabilities, and a selection channel awareness (SCA) based discriminator to counter SCA-based steganalysis[9]. Experimental results demonstrate that the proposed approach surpasses existing methods in terms of security performance and training time. Additionally, the paper showcases practical applications by utilizing syndrome trellis codes (STC) for secret message embedding and extraction.

In summary, our proposed scheme offers a promising avenue for secure steganography, bridging the gap between theoretical research and practical applications. By leveraging GANs and addressing critical challenges, we pave the way for robust and imperceptible secret communication. Future work will explore extensions to video and audio media, as well as adaptive embedding strategies.

The paper introduces UT-SCA-GAN, a novel steganography method based on generative adversarial networks (GANs). However, several research gaps remain. Firstly, the paper lacks a comprehensive comparison with other GAN-based steganography techniques beyond ASDL-GAN and S-UNIWARD. Evaluating UT- SCA-GAN against state-of-the-art methods like StegNet, HiDDeN, or SteganoGAN would enhance its credibility[9]. Secondly, the impact of individual components (U- Net generator, Tanh-simulator function, and SCA-based discriminator) on performance and security is not thoroughly analyzed. Ablation studies would clarify their significance. Lastly, the paper focuses solely on spatial image steganography, neglecting exploration into other domains (e.g., JPEG steganography, video steganography, or text steganography) and practical applications. A broader discussion on applicability and use cases would enrich the paper’s contribution.

### Invisible Steganography via Generative Adversarial Networks

The research paper introduces a novel Convolutional Neural Network (CNN) architecture called ISGAN specifically designed for image steganography. The primary objective of ISGAN[10] is to embed a secret grayscale image into a color cover image while maintaining both capacity and invisibility. Notably, the secret image is selectively embedded only in the Y channel of the cover image, preserving color information and minimizing perceptual impact. To enhance security, the paper leverages a generative adversarial network (GAN), which minimizes the divergence between the distributions of stego images and natural images. Additionally, the proposed scheme employs a mixed loss function based on Structural Similarity Index (SSIM)[10] and Multi-Scale SSIM (MS-SSIM). This combination ensures that the stego images are visually realistic and imperceptible to the human eye. Experimental evaluations conducted on various datasets, including LFW, PASCAL- VOC12, and ImageNet, demonstrate that ISGAN achieves state-of-the-art performance in terms of invisibility, security, and overall quality.

The research gap of this paper is the lack of a deep learning-based steganography method that can achieve high capacity, invisibility, and security simultaneously. The paper proposes a novel CNN architecture named ISGAN that can conceal a gray secret image into a color cover image and extract it exactly on the receiver side. The paper also introduces generative adversarial networks, a new steganography position, and a mixed loss function to improve the performance of the steganography task. The paper claims that ISGAN[10] can achieve state-of-the-art results on several datasets and outperform previous works.

# TABLE: Summary Of Research Papers

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Title** | **Year** | **Dataset** | **Model** | **Metric** | **Research Gap** |
| HIDING IMAGES IN PLAIN SIGHT: DEEP STEGANOGRAPHY [1] | 2017 | * ImageNet | CNN | * Number of intensity values off (out of 256) for each pixel, per channel, on cover and secret image   β Cover Secret   * Deep-Stego 0.75   2.8 3.6   * Deep-Stego 1.00   3.0 3.2   * Deep-Stego 1.25   6.4 2.8   * Cover Only 0.00   0.1 (n/a)  . | * Deep network steganog raphy: Hiding DNN   models in DNN models   * Video steganograph y: Hiding large data in video sequences |
| IMAGE STEGANOGRAPHY: A REVIEW OF THE RECENT ADVANCES [2] | 2021 | * Bossbase * Celeba * Tiny Image Net * Handwrit tenDigits | CNN GAN | * Uses PSNR,MSE,   SSIM,accuracy, BPP, and steganalysis accuracy.   * Measures quality, robustness, security, and capacity of steganography methods. | * Lack of benchmark datasets * Lack of benchmark datasets * Explora tion of GAN-   based method s |
| MULTI-IMAGE STEGANOGRAPHY USING DEEP NEURAL NETWORKS [3] | 2021 | * Tiny Image Net | DNN | * Sum of Squares Error(SSE) per pixel, per channelThe losses received for the below results after 750 epochs were as below - * Loss of Entire Setup   - 182053.70   * Loss secret1 - 51495.24 * Loss secret2 - 39911.16 * Loss secret3 - 39337.07 * Loss Cover - 51310.23 | * Lack of benchmark datasets * Limitati ons of traditio nal and CNN-   based method s   * Explora tion of GAN-   based method s |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| END-TO-END TRAINED CNN ENCODER- DECODER NETWORKS FOR IMAGE STEGANOGRAPHY [4] | 2018 | * IStego100 K * RGB- BMP   Steganal ysis Dataset | CNN | * Bits per pixel (BPP) = 8 * Peak Signal to Noise Ratio (PSNR) 32.9 db for encoder and 36.6 for decoder * Structural Similarity (SSIM) index of about 0.96 | * Balancing Trade-offs * Two Approaches * Optimal Results |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STEGANOGAN: HIGH CAPACITY IMAGE STEGANOGRAPHY WITH GANS [5] | 2019 | * COCO * DIV2K | CNN | * RS-BPP:   Div2K Dataset:   * Accuracy: 0.82 * RS-BPP: = 2.52 * PSNR: = 37.49 * SSIM: = 0.88 * COCO   Dataset:   * Accuracy: = 0.95 * RS-BPP: = 2.85 * PSNR: = 36.94 * SSIM: = 0.92 | * Constraints on cover and secret image size/type. * Inability to scale to high payloads. |
| AN OPTIMIZED STEGANOGRAPHY HIDING CAPACITY AND IMPERCEPTIBLY USING GENETIC ALGORITHMS [6] | 2019 | * USC-SIPI   image database   * BOSS   database   * BOWS2   database. | GAs LSB | * The authors used peak signal-to-noise ratio (PSNR) as the main metric to measure the quality of the stego images6. They also used histogram analysis and statistical tests to evaluate the security and robustness of   their scheme. | * most of the schemes focus on the embedding procedure without considering the proper position of embedded data * steganography can be viewed as   a search and an optimization problem, but few studies have used GAs to solve it. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| THE DEVELOPMENT AND NEW DIRECTION OF DIGITAL IMAGE STEGANOGRAPHY [7] | 2020 |  | GAN | * The paper uses two metrics to evaluate the performance of the steganographic models: steganographic security and image quality. Steganographic security is measured by the detection accuracy of the steganalysis methods, such as SRM, GNCNN, and SCA-GFR.   Image quality is  measured by the peak signal-to- noise ratio (PSNR) and the structural similarity index  (SSIM) of the stego images | * The traditional adaptive steganography methods are vulnerable to the steganalysis methods based on deep learning, especially when the embedding rate is high. * The existing steganographic methods based on GANs or adversarial examples have some limitations, such as low embedding capacity, unstable message extraction, or high embedding distortion. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| IMAGE STEGANOGRAPHY ANALYSIS BASED ON DEEP LEARNING [8] | 2020 | * BOSSBase 1.01 * BOWS2 | CNN | * The paper used two metrics to evaluate the performance of the models: accuracy and F1-score, which are commonly used measures for classification tasks. | * The paper identified several research gaps in the field of image steganalysis, such as the difficulty of detecting low embedding rate steganography, the lack of global statistical information in feature learning, and the challenge of handling multiple types of steganography algorithms. The paper aimed to address these gaps by proposing novel methods based on   deep learning and transfer learning. |
| SPATIAL IMAGE STEGANOGRAPHY BASED ON GENERATIVE ADVERSARIAL NETWORK [9] | 2018 | * SZUBase * BOSSBase- 1.01 | GAN | * Error rates. This is the metric used to evaluate the security performance of the proposed scheme and compare it with other steganographic methods. It is measured by the percentage of incorrectly classified   images by steganalysis methods such as SRM and maxSRMd2. | * The existing steganographic methods based on deep learning are still inferior to the conventional hand- crafted methods in terms of security performance and training time. The optimal embedding simulator cannot propagate gradient in the adversarial training process.   The current steganalysis methods based on SCA are hard to resist by the existing   * steganographic methods |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| INVISIBLE STEGANOGRAPHY VIA GENERATIVE ADVERSARIAL NETWORKS [10] | 2018 | * LFW * ImageNet | GAN | * SSIM   (Structural Similarity Index), PSNR (Peak Signal to Noise Ratio), CNN- based steganalysis model | * The paper addresses the limitations of previous deep learning based steganography methods, such as low capacity, color distortion, and weak security. * It proposes a novel CNN architecture that can hide a gray secret image into a color cover image with high invisibility and security. * It also introduces a mixed loss function based on SSIM and MS-SSIM to better   fit the human visual system |

**3 PROPOSED ARCHITECTURE**

﻿To create an image encryption and decryption machine using deep steganography, we use a multi-level technique. Initially, we generate a cover photograph making use of Stable Diffusion 2, a technique acknowledged for its robustness and stability in picture generation. This cover photo serves as the basis for concealing the name of the game records.

Subsequently, the cover photograph undergoes transformation within the Prep Network. Here, the name of the game photograph is prepared for embedding with the aid of converting color-based totally pixels into more informative capabilities, inclusive of edges. Additionally, if the secret picture is smaller than the duvet photo, the Prep Network increases its length, making sure compatibility for seamless integration.

Following the training phase, the Hiding Network comes into play. This network combines the converted secret picture from the Prep Network with the duvet image to generate the Container photograph. Through a series of convolution layers, the Hiding Network blends the name of the game picture into the quilt photograph, ensuring that the concealed statistics stays imperceptible to the human eye.

On the recipient's give up, the Reveal Network serves because the decoder, liable for extracting the hidden picture from the Container photo. Trained concurrently with the Hiding Network, the Reveal Network operates independently, using its discovered parameters to accurately reconstruct the secret photograph.

Throughout the schooling system, our system is optimized to limit the reconstruction blunders for both the quilt and mystery photographs. This entails finding a sensitive stability among maintaining the visible first-rate of the duvet photograph and making sure the secrecy and fidelity of the hid statistics. To achieve this, the networks are skilled on a diverse variety of herbal photographs sourced from the ImageNet database, allowing them to generalize correctly throughout numerous visible contexts.

In essence, our photo encryption and decryption machine leverages the synergy between the Prep, Hiding, and Reveal Networks to safely embed and extract secret information inside cowl pics, thereby providing a sturdy and effective answer for secure verbal exchange and statistics transmission.

**4 METHODOLOGY**

The Stable Diffusion 2 network, utilized in this model, is a sophisticated architecture designed for generating high-quality images. Built upon principles of diffusion models, it employs a series of iterative steps to refine input data, gradually enhancing its quality while maintaining stability throughout the process. In the context of this model, the Stable Diffusion 2 network serves as a crucial component for preparing inputs to the subsequent networks. By leveraging its capabilities, the model can effectively manipulate and enhance the input data, ensuring that it meets the desired criteria for further processing. This preparatory phase plays a vital role in optimizing the overall performance and output quality of the model, facilitating the generation of compelling and realistic visual content.

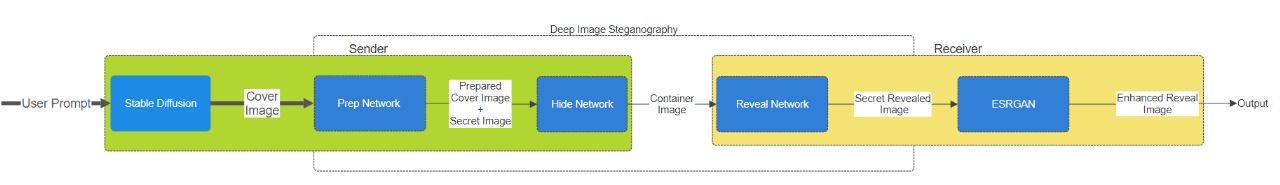


Figure 1: The five components of the full system. Stable Diffusion: Highlights the specific AI model for image generation. Prep Network: Secret-Image preparation. Hiding Network: Hiding the image in the cover image. Reveal Network: Uncovering the hidden image with the reveal network, ESRGAN: Emphasizes the final step of improving the extracted image quality.; this is trained simultaneously, but is used by the receiver.

**Prep Layer:**

At its core is the **PrepLayer**, which preprocesses the input images from stable diffusion and extracts features at different spatial resolutions. Subsequent layers further refine these features to facilitate downstream tasks such as image classification or object detection.

The **PrepLayer** serves as the foundation of the CNN architecture, performing initial processing and feature extraction. This layer consists of several key components:

Convolutional Layers for Different Kernel Sizes: The PrepLayer incorporates three convolutional layers (conv\_layer\_4\_3x3, conv\_layer\_4\_4x4, conv\_layer\_4\_5x5), each with four filters and distinct kernel sizes (3x3, 4x4, and 5x5). By employing convolutional operations with varying receptive fields, the network can capture spatial information at different scales.

Concatenation: Following the convolutional operations, the outputs of the three layers are concatenated along the channel axis using the Concatenate layer (concat\_1). This process combines features extracted at multiple scales into a single representation, enabling the network to leverage both local and global context.

Additional Convolutional Layers: To further refine the fused representation, three additional convolutional layers (conv\_1\_3x3, conv\_1\_4x4, conv\_1\_5x5) are applied. Each of these layers has a single filter and kernel size matching one of the previous convolutional operations. This configuration allows for fine-grained adjustments to the extracted features.

Final Concatenation: The outputs of the additional convolutional layers are concatenated once again along the channel axis (concat\_2), producing the final output of the PrepLayer. This consolidated feature map contains rich spatial information captured across multiple scales

Activation Function: Throughout the architecture, the Rectified Linear Unit (ReLU)

activation function is employed to introduce non-linearity and facilitate feature learning.

Input Normalization: Prior to processing, the input images are rescaled to a range of [0, 1] using the Rescaling layer. This normalization step ensures consistency in input values and aids in model convergence during training.

Training Parameter: The training parameter is utilized to control the behavior of certain layers, such as dropout and batch normalization, during training and inference phases.

**Hiding Network:**

It appears to be designed for image processing tasks, likely related to image hiding or steganography, given the presence of layers such as **HideLayer** and operations like concatenation and convolution. The network aims to take two input tensors, preprocess them, perform various convolutions, and produce an output tensor. This layer serves as the main component of the network architecture.

It takes two input tensors and processes them through several convolutional and concatenation operations.

The primary purpose seems to be to combine information from both input tensors and extract features using multiple convolutional filters.

Finally, it generates an output tensor representing the hidden information.

PrepLayer: A custom layer (PrepLayer) responsible for some preprocessing operations on the first input tensor.

Concatenation Layers:

Three concatenation layers (concat\_1, concat\_2, and concat\_3) are used to combine tensors at various stages of processing.

They concatenate tensors along the depth dimension (axis=3), likely to combine feature maps from different convolutional paths.

Multiple convolutional layers (conv\_layer\_4\_3x3, conv\_layer\_4\_4x4, conv\_layer\_4\_5x5, conv\_1\_3x3, conv\_1\_4x4, conv\_1\_5x5, conv\_1\_1x1) with different configurations (number of filters, kernel sizes) are used for feature extraction and transformation.

Activation functions ReLU (tf.nn.relu) are applied after each convolutional layer to introduce non-linearity.

A rescaling layer is applied to rescale the second input tensor. This layer ensures that the input values are within a certain range, which is a common preprocessing step in image-related tasks.

The network takes two input tensors.

The first tensor (prep\_input) undergoes preprocessing using the PrepLayer.

The second tensor (hide\_input) is rescaled to ensure its values are within a specific

range. The output tensor is produced by the conv\_1\_1x1 layer, representing the final result of the processing pipeline.

It likely represents the hidden information extracted from the input tensors.

**Reveal Network:**

The **RevealLayer** network consists of several convolutional layers followed by concatenation operations to merge feature maps. The key components of the architecture are as follows: Three convolutional layers are used to process the input tensor with different kernel sizes: 3x3, 4x4, and 5x5.

Each convolutional layer applies 50 filters and ReLU activation function to extract features from the input.

These convolutional layers operate independently on the input tensor, capturing different aspects of the input data.

After processing the input tensor through the three convolutional layers with different kernel sizes, their output feature maps are concatenated along the depth dimension (axis=3).

This concatenation operation combines the features extracted by the convolutional layers, allowing the network to capture diverse information from the input.

Following the concatenation operation, another set of convolutional layers is applied to the concatenated feature maps.

Three convolutional layers with kernel sizes 3x3, 4x4, and 5x5 are used, each with 50 filters and ReLU activation.

These additional convolutional layers further refine the features obtained from the concatenated feature maps.

After processing the input tensor through the three convolutional layers with different kernel sizes, their output feature maps are concatenated along the depth dimension (axis=3).

This concatenation operation combines the features extracted by the convolutional layers, allowing the network to capture diverse information from the input.

The final convolutional layer with a 1x1 kernel size and 3 filters is employed to reduce the depth of the feature maps and generate the final output.

This layer helps in transforming the concatenated feature maps into the desired output

format, which typically corresponds to the task at hand (e.g., classification, segmentation, etc.).The network is implemented as a custom layer in TensorFlow/Keras, inheriting from

the tf.keras.layers .Layer class.

Each convolutional layer and concatenation operation is defined within the \_init\_ method of the custom layer, ensuring proper initialization of network parameters. The call method is responsible for executing the forward pass of the network, where input tensors are processed through the defined layers and operations. During the forward pass, input tensors are sequentially passed through the convolutional layers, followed by concatenation operations, and finally through the output convolutional layer to produce the final output.

**Loss Function:**

The Steganography Loss network presented here is a custom loss function designed specifically for training steganography models. This loss function combines Mean Squared Error (MSE) terms for the secret message and cover data. By adjusting the weight of the MSE term for the secret message, controlled by the parameter beta, the network aims to find an optimal balance between imperceptibility and robustness of the embedded secret information. The Steganography Loss network is implemented as a subclass of tf.keras.losses. Loss, allowing it to seamlessly integrate into TensorFlow-based models.

The \_init\_ method initializes the loss function, allowing customization through the beta parameter, which determines the relative importance of the secret message MSE term compared to the cover data MSE term. The call method is the core of the loss function, where the actual computation takes place. It takes two inputs: y\_true and y\_pred, which represent the ground truth and predicted values, respectively, for both the secret message and cover data.

Inside the call method:It extracts the secret message and cover data from both y\_true and y\_pred. Calculates the MSE between the true and predicted values for both the secret message and cover data using TensorFlow's tf.losses.MSE. Combines the MSE terms, weighted by the beta parameter. Finally, computes the mean of the combined MSE terms, representing the overall loss value.

MSE is a common metric for measuring the difference between two sets of values. In this context, it quantifies the distortion introduced by the steganography process.By computing the MSE separately for the secret message and cover data, the loss function captures both the imperceptibility of the embedded secret and the fidelity of the cover data.

The beta parameter controls the trade-off between embedding capacity (controlled by the secret message MSE) and perceptual quality (controlled by the cover data MSE). Higher values of beta prioritize minimizing distortion in the secret message over imperceptibility in the cover data, and vice versa.

Taking the mean of the combined MSE terms ensures that the loss value is normalized, making it independent of the batch size and suitable for optimization algorithms that rely on gradient descent.

L(c, c0 , s, s0 ) = ||c − c 0 || + β||s − s 0 || (1)

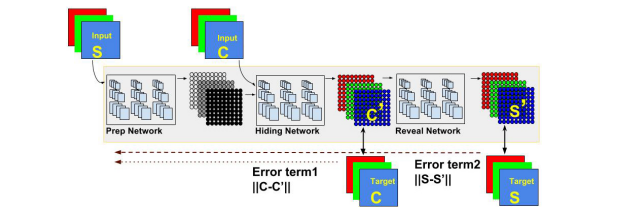


Figure 2: The three networks are trained as a single, large, network. Error term 1 affects only the first two networks. Error term 2

affects all 3. S is the secret image, C is the cover image.

**5 EXPERIMENTATION AND TESTING RESULTS**

Dataset:

We train our model on Tiny Imagenet dataset which has 50,000 images but we train our model on 6750 images. The images are 64x64 pixels.

We test our model on two datasets 50k celeb dataset 64x64 and cat faces dataset for gaining results.

Model Creation:

﻿In our schooling manner, we followed specific hyperparameters and configurations tailored to optimize the performance of our version. With a BATCH\_SIZE of 32 and EPOCHS set to seventy five, we aimed to strike a balance among computational efficiency and model convergence over the training iterations. Each epoch encompassed 210 iterations, facilitating an ample range of education updates to refine the version's parameters even as stopping overfitting.

Crucially, the choice of studying rate is pivotal in guiding the optimization method. By setting the LEARNING\_RATE to 1e-3, we aimed to make sure steady progress towards minimizing our custom loss characteristic. This price turned into selected via experimentation and cautious consideration of the model's convergence dynamics, making sure that the optimization manner neither stagnated nor diverged due to excessively massive or small updates.

Our custom loss featurezeros − s zero, encapsulates the essence of our project, which involves simultaneously reconstructing each the cover photo (c) and the name of the game picture (s) while minimizing the discrepancy between the reconstructed and authentic images (c0 and s0, respectively). Here, β, serving as a regularization parameter, was set to at least one, reflecting identical significance placed on constancy in reconstructing each the duvet and secret photos. This balance become deemed critical to make sure that neither element of the reconstruction manner become prioritized over the opposite, thereby preserving the integrity and safety of the hid information.

By meticulously configuring those hyperparameters and designing our loss feature to encapsulate the targets of our challenge, we aimed to equip our model with the essential gear to efficaciously research and generalize from the training facts, in the long run main to strong and reliable performance in actual-international situations

Model Summary:

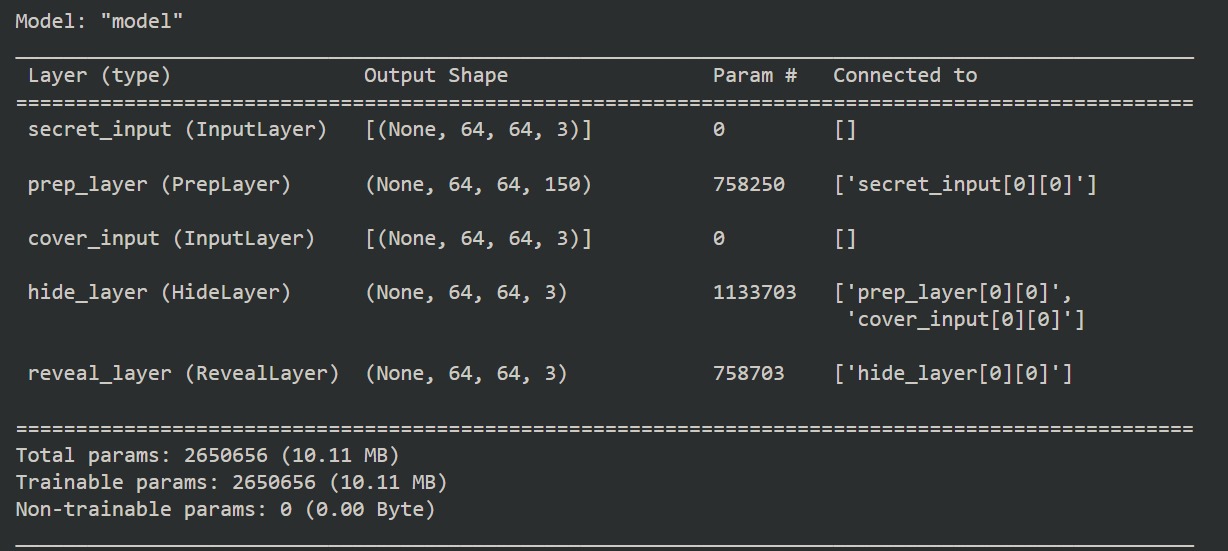


Figure 3: Model Summary, the figure shows trainable and non trainable parameters of our complete network for deep image steganography

Training Images for Epoch 1-20:

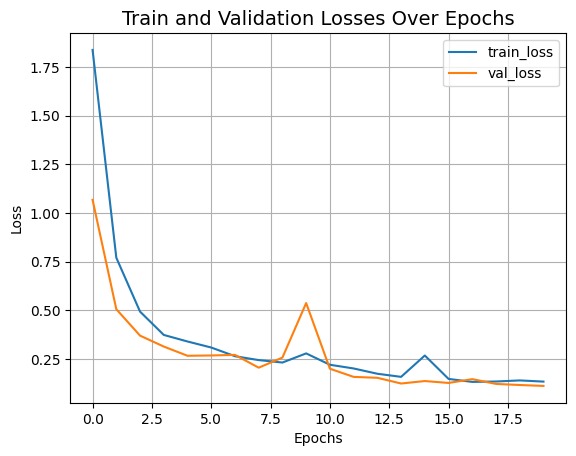


Figure 4. Train and Validation Losses Over Epochs

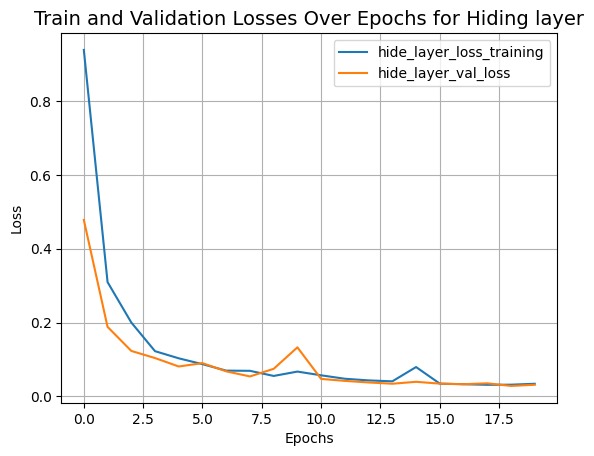


Figure 5. Train and Validation Losses Over Epochs for Hiding Layer

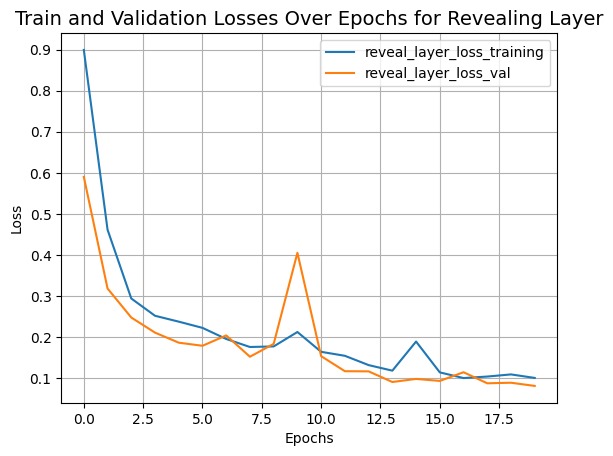


Figure 6. Train and Validation Losses Over Epochs for Revealing Layer

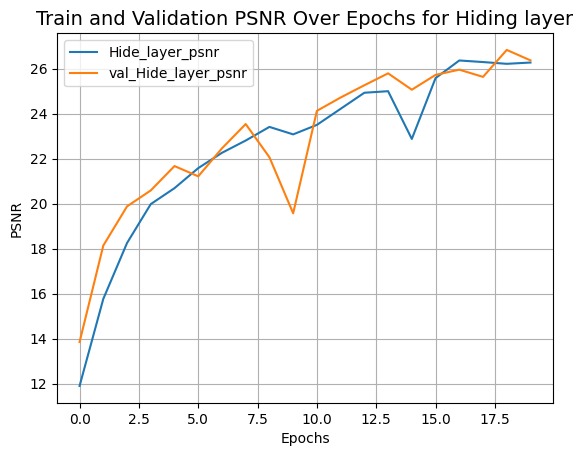


Figure 7. Train and Validation PSNR Over Epochs for Hiding Layer

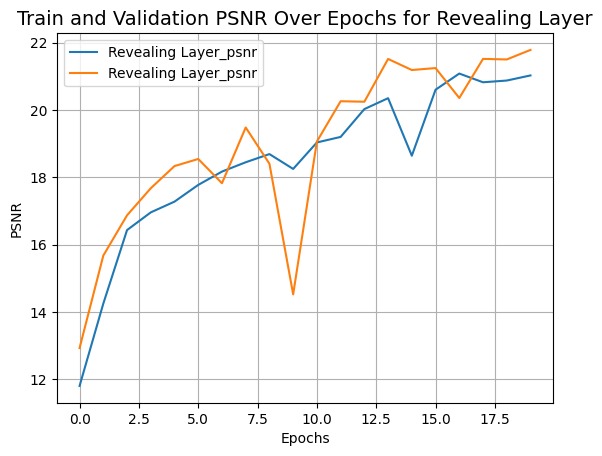


Figure 8. Train and Validation PSNR Over Epochs for Revealing Layer

Training Images for Epoch 20-75:

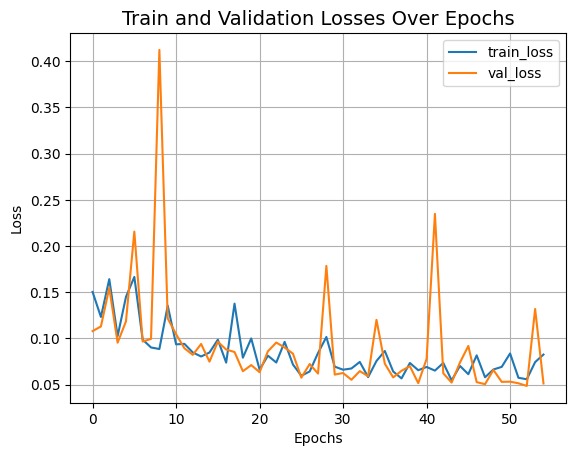


Figure 9. Train and Validation Losses Over Epochs

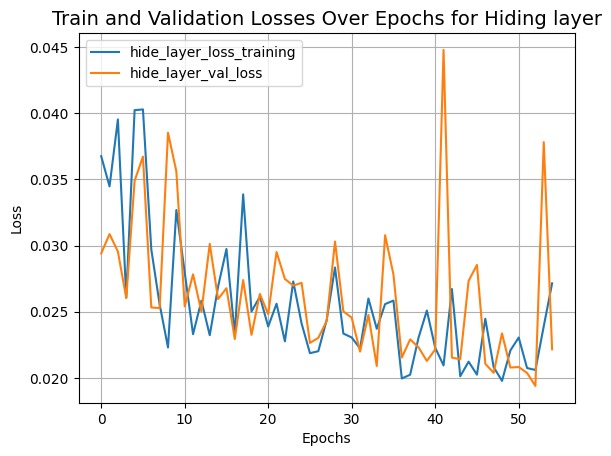


Figure 10. Train and Validation Losses Over Epochs for Hiding Layer

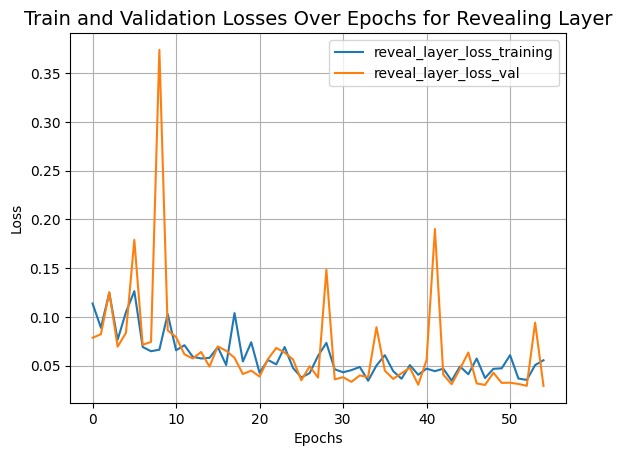


Figure 11. Train and Validation Losses Over Epochs for Revealing Layer

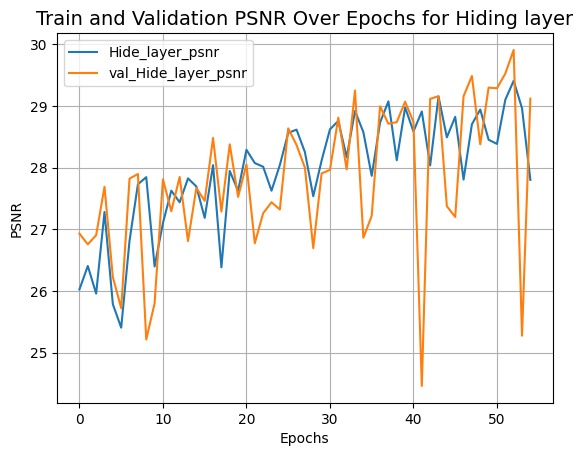


Figure 12. Train and Validation PSNR Over Epochs for Hiding Layer

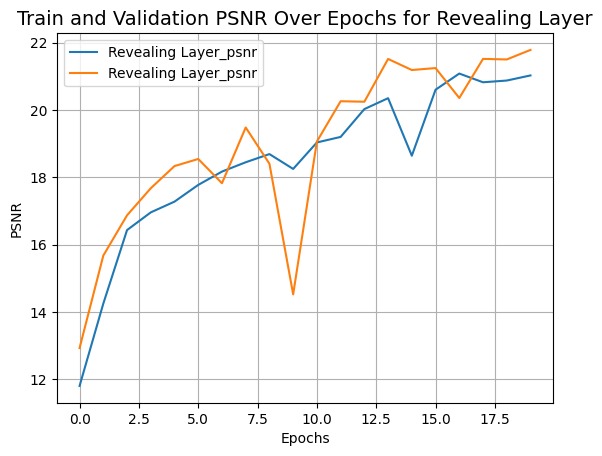


Figure 13. Train and Validation PSNR Over Epochs for Revealing Layer

**TESTING/RESULTS**

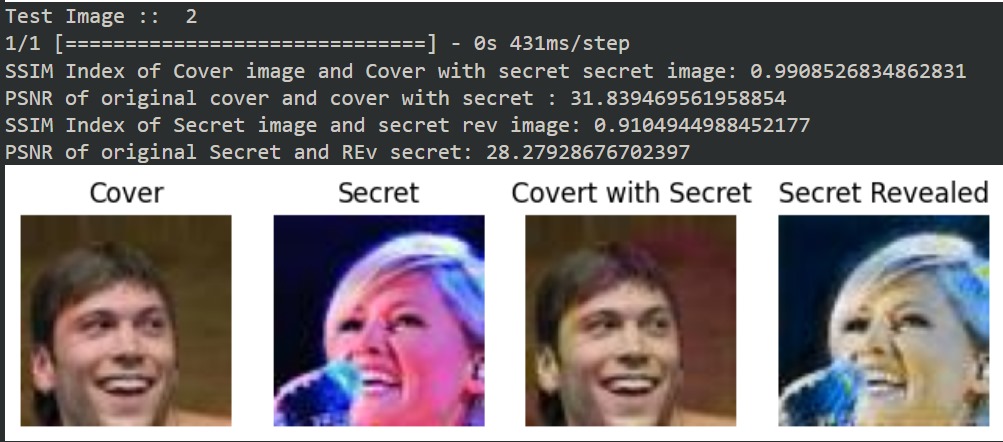


Figure 14. Test Image 1

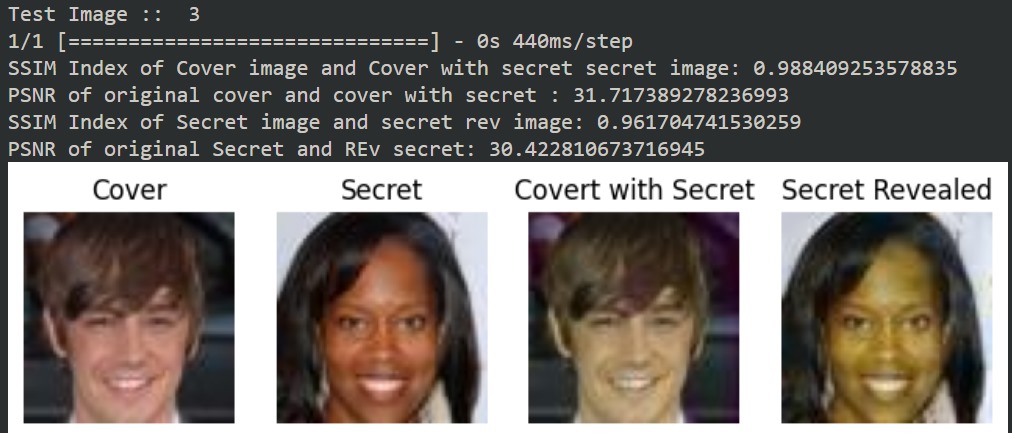


Figure 15. Test Image 2

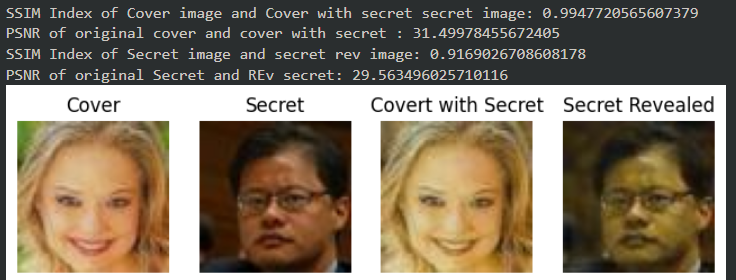


Figure 16. Test Image 3

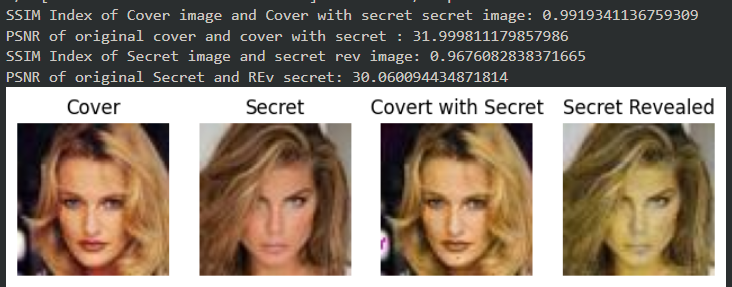


Figure 17. Test Image 4

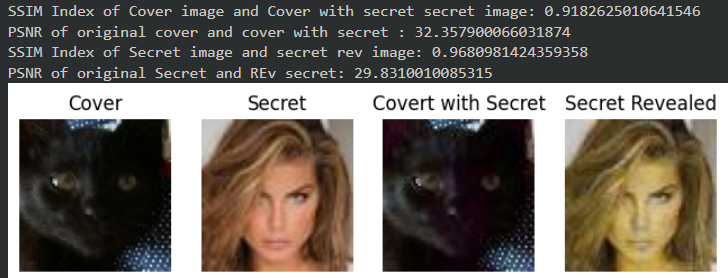


Figure 18. Test Image 5

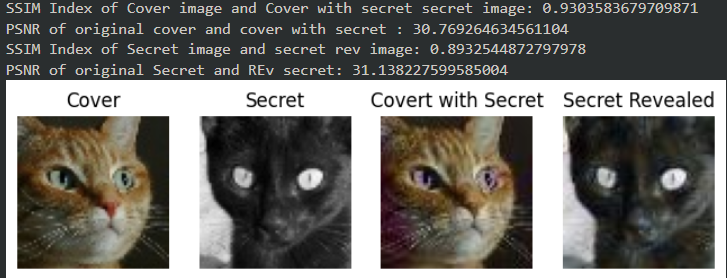


Figure 19. Test Image 6

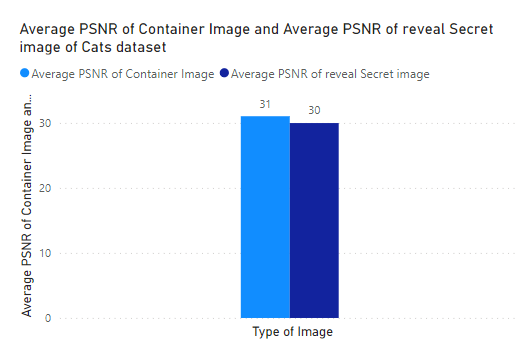


Figure 20. Average PSNR of Container Image and Average PSNR of reveal Secret image of Cats Dataset

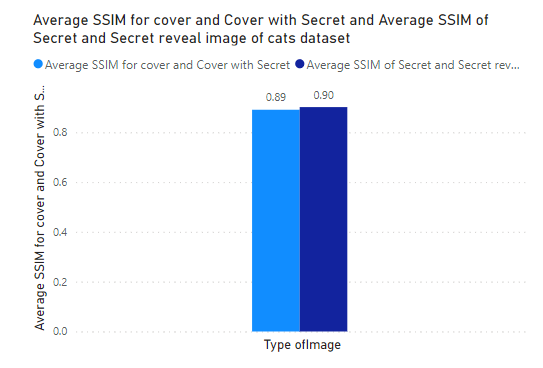


Figure 21. Average SSIM for cover and Cover with Secret and Average SSIM of Secret and Secret reveal image of Cats Dataset

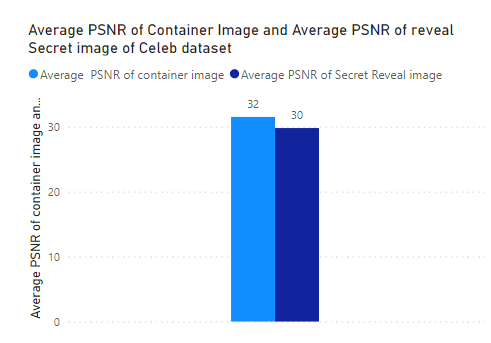


Figure 22.Average PSNR of Container Image and Average PSNR of reveal Secret image of Celeb Dataset

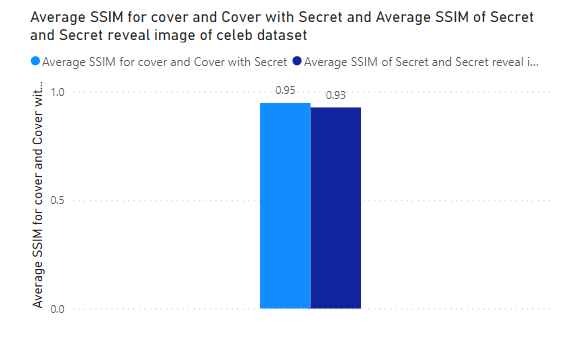


Figure 23.Average SSIM for cover and Cover with Secret and Average SSIM of Secret reveal image of Celeb Dataset

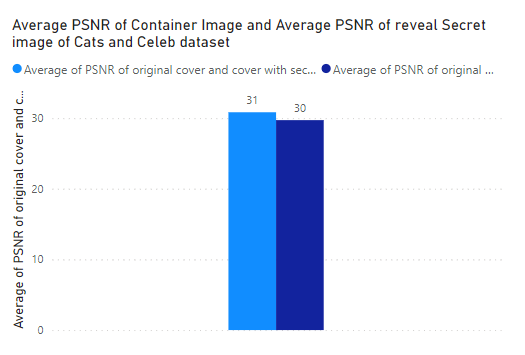


Figure 24.Average PSNR of Container Image and Average PSNR of reveal Secret image of Cats and Celeb Dataset

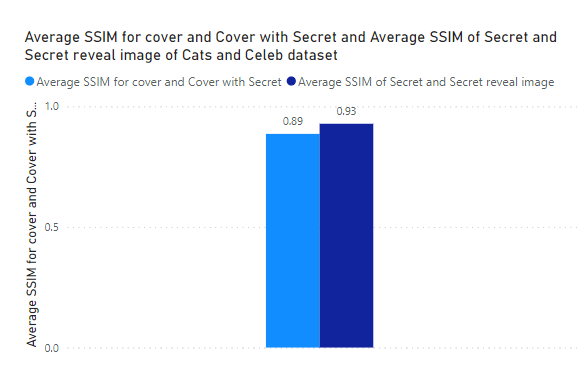


Figure 25.Average SSIM for cover and Cover with Secret and Average SSIM of Secret reveal image of Cats and Celeb Dataset

**6 CONCLUSION**

The preceding discussion and experimental findings illustrate the practical implementation of our deep image steganography system. Through showcased examples, it becomes evident that our model withstands decryption attempts using conventional techniques. Analysis of Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) values indicates commendable performance in its current state. Notably, as the original paper lacks performance metrics [1], our endeavor not only modularizes the system but also rigorously evaluates it under various conditions. Despite limitations imposed by GPU constraints, the conducted experiments offer valuable insights.

However, avenues for future refinement persist. Experimentation with parameters such as beta, utilization of high-resolution datasets, and model fine-tuning hold promise for enhancing output quality. These potential enhancements signify ongoing efforts towards optimizing the system's efficacy and resilience. Thus, while our current findings affirm the robustness of our approach, they also underscore the scope for continued research and improvement.

**REFERENCES**

1. Baluja, Shumeet. "Hiding images in plain sight: Deep steganography." Advances in neural information processing systems 30 (2017).
2. Subramanian, Nandhini, et al. "Image steganography: A review of the recent advances." IEEE access 9 (2021): 23409-23423.
3. Das, Abhishek, et al. "Multi-image steganography using deep neural networks." arXiv preprint arXiv:2101.00350 (2021).
4. Rahim, Rafia, and Shahroz Nadeem. "End-to-end trained cnn encoder- decoder networks for image steganography." Proceedings of the European conference on computer vision (ECCV) workshops. 2018.
5. Zhang, Kevin Alex, et al. "SteganoGAN: High capacity image steganography with GANs." arXiv preprint arXiv:1901.03892 (2019).
6. Wazirali, Ranyiah, et al. "An optimized steganography hiding capacity and imperceptibly using genetic algorithms." IEEE Access 7 (2019): 133496- 133508.
7. Huang, He, et al. "The Development and New Direction of Digital Image Steganography." 2020 International Conference on Robots & Intelligent System (ICRIS). IEEE, 2020.
8. Kumar, Vijay, Pankaj Rao, and Ashish Choudhary. "Image steganography analysis based on deep learning." Review of Computer Engineering Studies 7.1 (2020): 1-5
9. Yang, Jianhua, et al. "Spatial image steganography based on generative adversarial network." arXiv preprint arXiv:1804.07939 (2018).

[10]Zhang, Ru, Shiqi Dong, and Jianyi Liu. "Invisible steganography via generative adversarial networks." Multimedia tools and applications 78 (2019): 8559-8575.