Neural Driven embedding in images for secure Image Steganography

A M. Tech Project Report Submitted
in Fulfillment of the Requirements
for the Degree of

Master of Technology

by

Shardul Nalode
(234101049)

under the guidance of

Prof. Pinaki Mitra

to the

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI
GUWAHATI - 781039, ASSAM

CERTIFICATE

This is to certify that the work contained in this thesis entitled "**Neural Driven embedding in images for secure Image Steganography**" is a bonafide work of **Shardul Nalode** (**234101049**) , carried out in the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati under my supervision and that it has not been submitted elsewhere for a degree.

Supervisor: **Prof. Pinaki Mitra**

Professor,

June, 2025 Department of Computer Science & Engineering,

Guwahati. Indian Institute of Technology Guwahati, Assam.

Acknowledgements

We want to express our thanks, to Professor **Pinaki Mitra** for his advice and support. and constant encouragement, during our M.Tech Thesis Project has been invaluable, to us. Our team's guidance and support have played a role, in steering the direction of our research endeavors. We truly appreciate it. Grateful, for the wisdom and guidance he kindly shared with us. This has been truly inspiring. Engaging in pursuits is truly a journey.

Abstract

In recent years, the need for secure and covert communication has grown significantly, especially with the increasing exchange of digital media. This thesis explores a deep learning-based method for image steganography, where a complete image is embedded inside another image using an end-to-end neural network framework. Unlike traditional steganographic approaches that focus on hiding text or binary data, this work aims to conceal high-resolution visual content, making the task more challenging and realistic.

The proposed method introduces a neural architecture composed of three sequential networks: a Prep Network that extracts essential features from the secret image, a Hiding Network that embeds those features into the cover image to generate a stego image, and a Reveal Network that reconstructs the secret image from the stego image. These networks are trained jointly using a composite loss function that combines pixel-level differences, structural similarity, and perceptual quality to ensure both the invisibility of the stego image and the fidelity of the recovered image.

Comprehensive experiments were conducted on standard datasets to assess the performance of the system under various conditions. The quality of the stego image was evaluated using metrics like PSNR, SSIM, and LPIPS, ensuring minimal distortion compared to the original cover. Simultaneously, the accuracy of the recovered secret image was also measured across the same metrics. Results show that the model can effectively hide and retrieve images without noticeable degradation, making it suitable for practical use in scenarios where visual integrity and confidentiality are essential.

This work establishes a strong baseline for secure image-to-image steganography using neural networks and lays the foundation for future improvements involving attention mechanisms, multi-scale feature fusion, and adversarial training to further enhance both robustness and invisibility.

Contents

List of Figures

List of Tables

+

Chapter 1

Introduction

1.1 What is Steganography?

Steganography is the practice of concealing information within other seemingly harmless data such that the presence of the hidden content remains undetectable. Unlike cryptography, which secures the content but reveals the existence of a message, steganography masks the fact that any message exists at all. Historically, methods of steganography ranged from invisible ink to microdots, but in the digital era, it is predominantly applied to multimedia files—such as images, audio, and video.

Among these, images have emerged as the most widely used carriers for hidden information, owing to their large data capacity and the human eye's tolerance for small color variations. Minor changes to pixel values in an image can encode data without visibly altering the image, making visual steganography both effective and discreet.

However, modern use cases have exposed the limitations of traditional, rule-based embedding techniques. Classical approaches often rely on simple bit-level modifications, such as Least Significant Bit (LSB) substitution, which are vulnerable to statistical attacks and offer limited robustness to image transformations like compression, resizing, or noise.[1]

With growing demands for secure, high-capacity, and robust data hiding mechanisms, the focus has shifted to deep learning-based approaches that learn complex mappings between cover and secret data. These models offer greater flexibility and resilience, enabling image-to-image steganography, where an entire image can be embedded and later retrieved with high fidelity.

This thesis builds on that paradigm, aiming to develop a neural network-based framework that can hide one image inside another without perceptible loss in quality, while ensuring accurate recovery of the embedded content.

1.2 Problem Motivation

With the rapid expansion of digital communication, the need for secure and covert methods of transmitting sensitive data has become more critical than ever. Traditional steganographic techniques, while simple and lightweight, often suffer from limited capacity, poor robustness against image transformations, and vulnerability to steganalysis attacks. Embedding binary or textual data using fixed rules like LSB manipulation can be easily detected or disrupted, especially when images are compressed, resized, or modified in transit.

This motivates the shift towards neural-driven steganography, which enables end-to-end learning of complex mappings between cover and secret images. By leveraging the power of convolutional neural networks, it is possible to embed high-dimensional visual content in a way that is both visually imperceptible and structurally recoverable.

The core objective of this work is to design and evaluate a deep learning framework that can securely embed a complete image within another image while preserving the visual quality of the cover and ensuring accurate reconstruction of the hidden image. This approach moves beyond traditional binary embedding and focuses on learning robust, high-capacity representations that align with modern security and data integrity demands.

The ultimate goal is to contribute to the development of steganographic systems that can be used for privacy-preserving visual communication, digital watermarking, and secure media sharing in environments where data integrity and stealth are equally critical.

1.3 Our contributions

- **Methodology:**

  This work proposes a deep learning-based framework for secure image-to-image steganography, aimed at embedding one full-resolution image into another without perceptible distortion. The architecture is divided into three neural modules: a *Prep Network* that extracts essential features from the secret image, a *Hiding Network* that blends these features into the cover image to generate a stego image, and a *Reveal Network* that reconstructs the hidden image from the stego counterpart.

  Training is performed end-to-end using a combination of losses — including Mean Squared Error (MSE), Structural Similarity Index (SSIM), and perceptual metrics — to balance the fidelity of both the stego and revealed images. The model ensures that the embedded data remains robust while the visual quality of the cover image is preserved. To further improve performance, architectural enhancements such as skip connections, residual layers, and multi-scale feature learning are also explored.

  The system is evaluated on publicly available imagenet datasets to assess embedding quality, recovery accuracy, and robustness under common image distortions like compression and resizing. Quantitative evaluation is done using PSNR, SSIM, and LPIPS to validate the reliability and invisibility of the method.

- **Real-World Applicability:**

  The proposed technique holds promise in several real-world use cases where visual security and covert communication are essential. It can be used in secure messaging systems to transmit sensitive images without revealing the presence of hidden data. In digital watermarking, intellectual property rights can be protected by embedding unique ownership patterns directly into images without altering their visual content.

Moreover, the method enhances privacy in cloud storage environments by embedding metadata or personal information within innocuous-looking images, making unauthorized access significantly more difficult. Law enforcement, military communication, and healthcare sectors can also benefit from this approach, enabling confidential transfer of visual data without depending entirely on encryption, which might attract attention or be subject to legal restrictions in certain contexts.

Additionally, this technique can be used for secure content distribution on social media platforms, where embedded authentication information or hidden messages can verify content legitimacy without affecting appearance. It can also assist in digital forensics and tamper detection by embedding invisible signatures or integrity checks into image files, enabling future verification of authenticity and detection of unauthorized modifications.

1.4 Summary of the Work

The objective of this work is to design and evaluate a deep learning-based framework for secure image-to-image steganography. Unlike traditional techniques that embed text or binary data, the proposed method focuses on hiding an entire secret image inside a cover image using convolutional neural networks. The main challenge is to ensure that the stego image is visually indistinguishable from the cover image, while the secret image can be recovered with minimal loss in quality.

To achieve this, the framework is composed of three interconnected neural networks: a Preparation Network that extracts high-level feature representations from the secret image, a Hiding Network that merges these features into the cover image to generate a stego image, and a Reveal Network that reconstructs the hidden image from the stego output. The entire system is trained in an end-to-end manner using a combination of pixel-level loss, structural similarity, and perceptual loss functions to ensure both invisibility and recoverability.

The model was evaluated using standard image datasets and validated across multiple metrics including PSNR, SSIM, and LPIPS. Experiments also explored architectural enhancements such as skip connections and residual layers to improve robustness and reconstruction quality. The results demonstrate that the system effectively embeds and recovers images without introducing significant artifacts or distortions, and can withstand common image transformations like resizing or mild compression.

The proposed system offers strong potential in applications like private media sharing, digital watermarking, and forensic image tracking. By hiding visual information within seemingly ordinary images, the framework enables secure communication and data protection without raising suspicion, making it suitable for use in environments where conventional encryption might be impractical or easily detectable.

Chapter 2

Literature Review

2.1 Overview of Steganography Techniques

Steganography is the science of hiding secret data within a non-suspicious carrier medium in such a way that the presence of the hidden content remains undetected. The objective is not just to protect the content, as in cryptography, but to obscure the fact that any communication is taking place. In digital contexts, this technique is commonly applied to multimedia files such as images, audio, and video.

Among these, images are the most popular carriers due to their high data capacity and the human visual system's tolerance to minor changes in pixel values. Image steganography typically operates by modifying specific aspects of an image, either in the spatial domain (e.g., pixel-level changes) or the transform domain (e.g., frequency-based modifications using transforms like DCT or DWT). The goal is to ensure that the modifications do not introduce perceptible artifacts while still allowing accurate retrieval of the hidden message.

The success of a steganographic system is generally measured by three key criteria: **imperceptibility**, **payload capacity**, and **robustness**. Imperceptibility ensures that the stego image appears visually identical to the cover image. Payload capacity defines how much information can be embedded without degrading quality. Robustness refers to the system's ability to preserve hidden data even after common image transformations like compression, resizing, or noise addition.

Over the years, steganographic methods have evolved from simple rule-based approaches to more complex, adaptive systems. Traditional methods like Least Significant Bit (LSB) substitution and transform-based embedding have been widely studied, but they often fall short in resisting detection and distortions. This has led to a growing interest in learning-based methods, especially deep learning, which can learn optimal embedding strategies from data and offer greater flexibility, resilience, and performance.

2.2 Traditional and Modern Approaches to Image Steganography

Traditional image steganography techniques rely on direct manipulation of pixel values or transform coefficients to hide data in cover images. These methods are typically lightweight and easy to implement, but often fall short in terms of robustness and capacity. Over time, deep learning has introduced more flexible, secure, and powerful approaches. This section describes both traditional and learning-based techniques.

1. **Least Significant Bit (LSB) Substitution:** The Least Significant Bit technique is one of the earliest and simplest methods used for image steganography. It involves replacing the least significant bits of the pixel intensity values with the bits of the secret message. Since the human eye is relatively insensitive to small changes in color or brightness, such modifications usually go unnoticed. For an 8-bit grayscale image, altering the LSB changes the pixel value by at most one unit, which is visually negligible. Despite its simplicity and high embedding capacity, LSB substitution suffers from serious drawbacks. It is highly sensitive to image processing operations such as compression, cropping, or resizing, which can destroy or distort the embedded message. Furthermore, it is vulnerable to statistical steganalysis techniques that can detect patterns in pixel distributions introduced by uniform LSB manipulation.[1][2]

2. **Pixel Value Differencing (PVD):** Pixel Value Differencing improves upon the basic

LSB method by exploiting the local characteristics of the image. It uses the difference between adjacent pixel values to determine the embedding capacity for each region. In textured or edge-rich areas, where pixel differences are large, more bits can be embedded without noticeable artifacts. In contrast, smooth areas receive fewer embedded bits to maintain visual quality. This adaptive strategy offers a better trade-off between imperceptibility and capacity. However, PVD methods still operate in the spatial domain and are therefore susceptible to noise, filtering, and geometric transformations. Also, improper embedding in smooth regions may still result in detectable artifacts when analyzed statistically.[3]

3. **Discrete Cosine Transform (DCT) Embedding:** DCT-based steganography operates in the frequency domain and is particularly useful when the stego image is intended to be saved in JPEG format. The image is divided into blocks, and each block is transformed using the DCT to produce frequency coefficients. Secret data is embedded by modifying selected mid-frequency DCT coefficients to avoid visible distortion and maintain robustness against compression. Low-frequency components are avoided to prevent noticeable artifacts, while high-frequency components may be discarded during compression. This method offers better resilience to JPEG compression and signal degradation than spatial-domain techniques. However, selecting the right coefficients and embedding strength is critical—improper tuning can lead to detectable image degradation or poor recovery of hidden data.[4]

4. **Discrete Wavelet Transform (DWT) Embedding:** DWT is another frequency-domain technique that transforms an image into multiple subbands corresponding to different frequency and spatial components. The subbands (LL, LH, HL, and HH) represent approximation and detail information at various levels. Typically, data is embedded into the higher-frequency subbands (LH, HL, HH) where changes are less perceptible. Compared to DCT, DWT provides better spatial localization and multi-

resolution analysis, making it more flexible and effective for image steganography. Additionally, DWT-based methods are less sensitive to compression and allow for adaptive embedding across image scales. The downside is the increased computational complexity and the need for accurate inverse transformation to reconstruct the stego image.[5]

5. **F5 Algorithm and Matrix Encoding:** The F5 algorithm is a well-known JPEG steganographic technique that modifies DCT coefficients in a more efficient and less detectable way. It uses matrix encoding to minimize the number of changes needed for embedding, thereby reducing the distortion introduced into the cover image. It also employs permutative straddling to disperse embedded data randomly throughout the image, making it harder to detect via statistical analysis. F5 introduces a technique called ”shrinkage,” where coefficients are modified in a way that may result in zero values, reducing the chance of histogram artifacts. Although F5 is more secure than LSB and other spatial methods, it can still be exposed by advanced statistical analysis and is less suitable for formats other than JPEG.[6]

6. **HEVC-Based Steganography:** With greater efficiency and quality than its predecessors like H.264/AVC, HEVC (High-Efficiency Video Coding) has emerged as a leading standard in video compression. With features like variable quad-tree partitioning and intra-prediction modes (IPMs), this advanced coding standard has created new opportunities for safe data embedding in video streams. Unlike conventional picture steganography, video steganography in particular benefits from HEVC's design, which offers greater embedding capacities and reduces visual quality loss.[7]

   In steganography using HEVC technology as a base framework choice of cover material is influenced by Intra Prediction Modes (IPMs) for concealing information. choosing IPMs carefully to enhance security and minimize the risk of being detected by others. embedding is accomplished through coding unit (CU)size and statistical distributions.Maintaining distortion ensures the video quality remains intact and effective.

In attempt to to keep things under wraps,Matrix encoding is frequently used to boost how well information is hidden within data. needing adjustments, to the Intellectual Property Management systems that also assist in preserving the integrity of the video. The quality of visuals.

Evaluations of performance demonstrate that HEVC-based steganography techniques can achieve great data embedding efficiency with negligible effects on perceptual similarity and bit rate. According to metrics like BD-Bitrate (BDBR) and BD-SSIM, this method preserves video quality in a range of payload circumstances. Furthermore, those methods show a high level of resistance to steganalysis techniques based on IPM, which makes HEVC a reliable platform for safe data concealment in practical applications.[7]

In conclusion, HEVC-based steganography strikes a balance between low detectability and great embedding capacity by utilizing the special coding and compression capabilities of HEVC. For applications where maintaining video quality and guaranteeing security are crucial, this makes it a promising method for securely embedding video data.

7. **SteganoGAN for Image Steganography:** SteganoGAN introduces an innovative approach to hiding data within images using generative adversarial networks (GANs). This technique allows arbitrary binary data to be embedded seamlessly, ensuring the images remain visually natural and secure. Unlike traditional steganography, which is typically constrained to embedding about 0.4 bits per pixel and often leaves detectable traces, SteganoGAN pushes the boundaries by accommodating up to 4.4 bits per pixel. It achieves this through dense network connections and adversarial training, which enhance the stability and efficiency of the process while producing realistic and undetectable image modifications. With its encoder-decoder design, SteganoGAN can efficiently embed and retrieve data, while a critic network ensures the images maintain

their quality and look natural, making it a powerful advancement in the field.[8]

In terms of evaluation, SteganoGAN performs remarkably well. It maintains high peak signal-to-noise ratios (PSNR) and structural similarity index (SSIM) values even at maximum embedding capacities, achieving a PSNR of over 30 and SSIM values close to 1 in various scenarios. Furthermore, it successfully evades detection by traditional steganalysis tools, such as StegExpose, as well as more advanced, neural-based detection models, with low detection rates (around 0.6 area under the ROC curve). This adaptability, coupled with its high payload and minimal visual artifacts, demonstrates SteganoGAN's potential as a secure and efficient steganographic technique for embedding large amounts of data in images, making it an essential advancement in the field of steganography.[8]

8. **Neural Network-Based Steganography:** Neural networks have demonstrated significant potential in enhancing image steganography, providing a more sophisticated and secure approach to embedding data within cover images. Recent neural network-based methods leverage convolutional neural networks (CNNs) to effectively distribute hidden information across the pixels of the cover image, resulting in visually indistinguishable "container" images that make detection challenging for conventional steganalysis tools.[9] [10]

   The neural network-based steganography is a sophisticated and dependable method for safe image-based data hiding since it not only increases the capacity for data embedding but also guarantees high fidelity and resilience against steganalysis and image distortions.

2.3 Deep Learning-Based Steganography

The rise of deep learning has significantly transformed the landscape of image steganography by enabling models to learn optimal embedding patterns directly from data, without relying on handcrafted rules. Convolutional neural networks (CNNs), in particular, have been used to encode and decode hidden data in a visually imperceptible manner. These networks are capable of capturing high-level image semantics and spatial dependencies, making them suitable for complex steganographic tasks.

Several models based on encoder-decoder architectures have been developed for this purpose. In such frameworks, the encoder learns to embed a secret into a cover image, while the decoder is trained to extract it.

Despite these advancements, several limitations persist in the current state-of-the-art:

- **GAN-Based Approaches:** While generative adversarial networks (GANs), such as SteganoGAN, offer high payload capacity and realistic stego images, they come with substantial training complexity. Adversarial training is inherently unstable, often requiring careful hyperparameter tuning and large-scale datasets. Moreover, GANs tend to be resource-intensive, making them unsuitable for lightweight or real-time applications.[8]

- **HEVC-Based Techniques:** HEVC steganography methods leverage coding structures like intra prediction modes and coding unit partitioning. Although these techniques achieve high embedding capacity in video streams, they are tightly coupled with video compression standards and thus less effective for static image steganography. Their dependence on codec structure limits generalization and portability.[7]

- **Rule-Based Traditional Methods:** Classical techniques such as LSB substitution, DCT embedding, and pixel value differencing are easy to implement but highly vulnerable to compression, geometric attacks, and steganalysis. They also lack the adaptability required to handle diverse image content or varying noise conditions.

- **Lack of Robustness and Generalization:** Many deep learning-based models focus on ideal test cases and may not generalize well to real-world distortions such as JPEG compression, scaling, or color shifts. Embedding techniques that perform well on clean datasets can degrade significantly under practical conditions.

These challenges motivate the development of a more flexible, stable, and efficient deep learning architecture capable of embedding and recovering full-resolution images with high fidelity, minimal distortion, and strong robustness against steganalysis. Our proposed system builds upon these goals and is introduced in detail in the following chapter.

2.4 Summary

This chapter reviewed the evolution of image steganography from traditional rule-based techniques to modern deep learning-based approaches. Early spatial and frequency domain methods, such as Least Significant Bit (LSB) substitution, Pixel Value Differencing (PVD), and Discrete Cosine Transform (DCT), provided basic mechanisms for hiding information in images. While effective to some extent, these techniques were limited by low robustness, fixed embedding rules, and vulnerability to statistical steganalysis.

We also explored more advanced approaches, including HEVC-based steganography and deep learning methods like SteganoGAN. HEVC-based techniques offer high embedding capacity in video streams by exploiting intra prediction modes and coding unit structures. However, they are more suited for video data and less applicable for static image scenarios. GAN-based models, such as SteganoGAN, introduce adversarial training to produce highly imperceptible stego images with large payloads. Despite their strong performance, GANs require extensive computational resources and careful tuning, which may limit their practical deployment.

In contrast, neural network architectures based on convolutional encoders and decoders offer a balanced trade-off between embedding capacity, visual quality, and implementation complexity. These models can be trained end-to-end to learn optimal feature mappings for embedding and retrieving hidden images. Unlike GANs, they avoid adversarial instability and are easier to control, making them well-suited for real-world use cases where robustness, scalability, and training stability are essential.

Based on these observations, our work focuses on a neural-driven image-to-image steganography framework, leveraging a three-part architecture: Prep Network, Hiding Network, and Reveal Network. This system enables the secure embedding of one image within another, with high fidelity, minimal distortion, and strong resistance to steganalysis.

Chapter 3

Neural Driven embedding in images for secure Image Steganography

3.1 Proposed Work

This thesis proposes a deep neural network-based framework for secure image steganography, where a secret image is embedded into a cover image such that the resulting stego image remains visually indistinguishable from the original. The framework draws inspiration from image-to-image translation models and is designed with a focus on imperceptibility, robustness, and accurate reconstruction.

The core of the proposed method is a three-stage neural architecture trained end-to-end. The pipeline consists of a preparation stage to extract rich features from the secret image, a hiding stage that embeds these features into the cover image, and a reveal stage that reconstructs the secret from the stego image. The architecture leverages GELU activations and residual connections to enhance representational power and training stability, while multi-kernel convolutions capture both fine and coarse details.

To guide the training process, a custom loss function is employed that combines Mean Squared Error (MSE) and Structural Similarity Index Measure (SSIM), weighted by hyperparameters α and β. This hybrid loss ensures a balance between pixel-wise fidelity and perceptual quality for both the stego and revealed secret images.

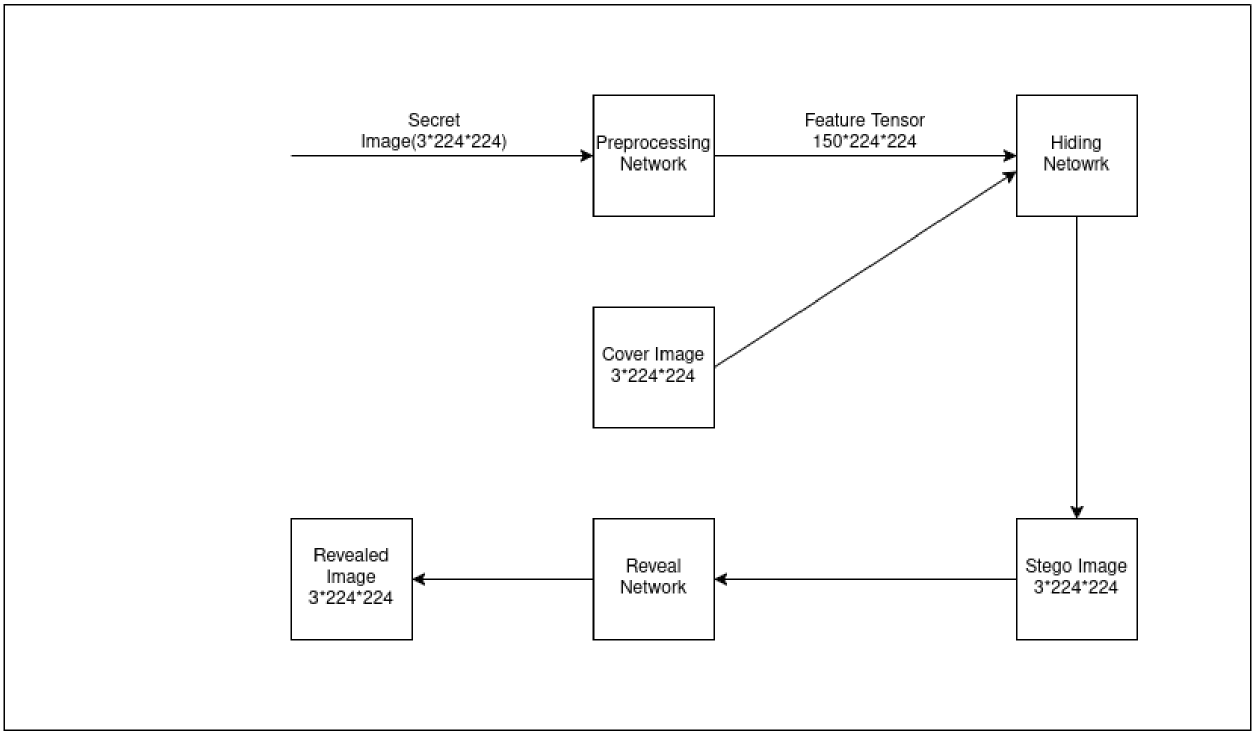


Fig. 3.1: Flow chart for proposed method.

3.2 Overall Approach

3.2.1 Network Architecture

The proposed architecture follows a three-stage design: *Preparation*, *Hiding*, and *Reveal* networks, implemented via a modular deep convolutional framework in PyTorch. The key innovations lie in the use of GELU activations, multi-kernel streams, and carefully designed residual blocks to enhance feature learning and stability.

Preparation Network: The `PrepNetwork` processes the secret image through three parallel convolutional streams with kernel sizes 3×3, 4×4, and 5×5. Each stream consists of 4 convolutional layers with GELU activations. The outputs of all streams are concatenated and passed through a second stage of convolutional layers, again with varying kernels, to yield a combined secret feature tensor.

Workflow: The secret image is fed into all three streams simultaneously, where each stream captures features at different receptive fields. After local and contextual features are extracted, the outputs are concatenated depth-wise. This concatenated tensor undergoes another set of convolutions to further fuse and refine the multi-scale information. The final output is a robust, high-dimensional feature map that represents the secret image in a form suitable for embedding.

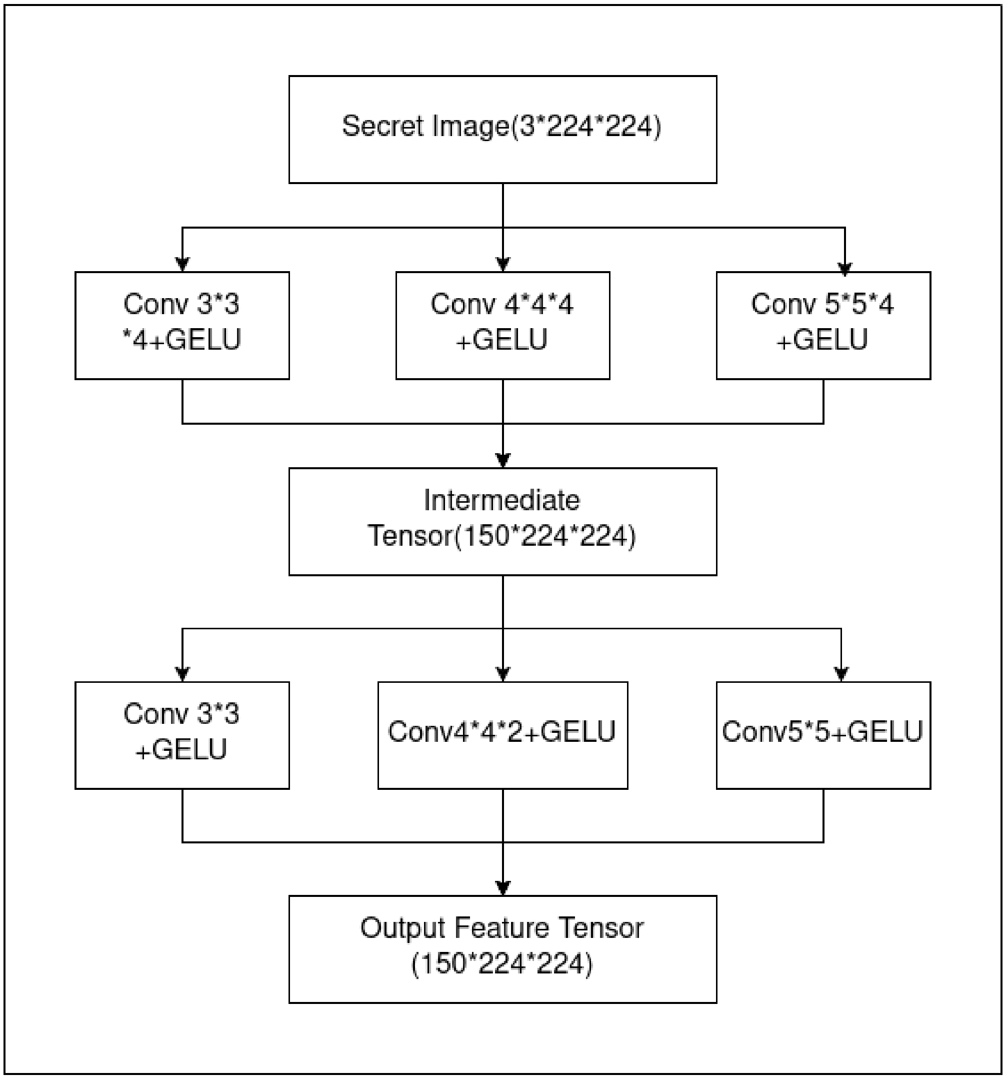


Fig. 3.2: Flow chart for Prep Network.

Hiding Network: The `HidingNetwork` takes the concatenation of the prepared secret features and the original cover image. It uses three convolutional branches (3×3, 4×4, 5×5), each enhanced with two *Residual Blocks* placed after every two convolutions. This design helps retain feature integrity while facilitating gradient flow during training. The outputs are resized to a common shape and merged to produce the final stego image.

Workflow: The concatenated tensor (secret + cover) is processed in parallel by the three branches, each extracting features at different granularities. Within each branch, residual connections ensure stable learning and help preserve the spatial information of both inputs. After passing through their respective layers, all branch outputs are interpolated to a unified spatial resolution. These are then concatenated and passed through final convolutional layers to produce a visually indistinguishable stego image that embeds the secret information securely.

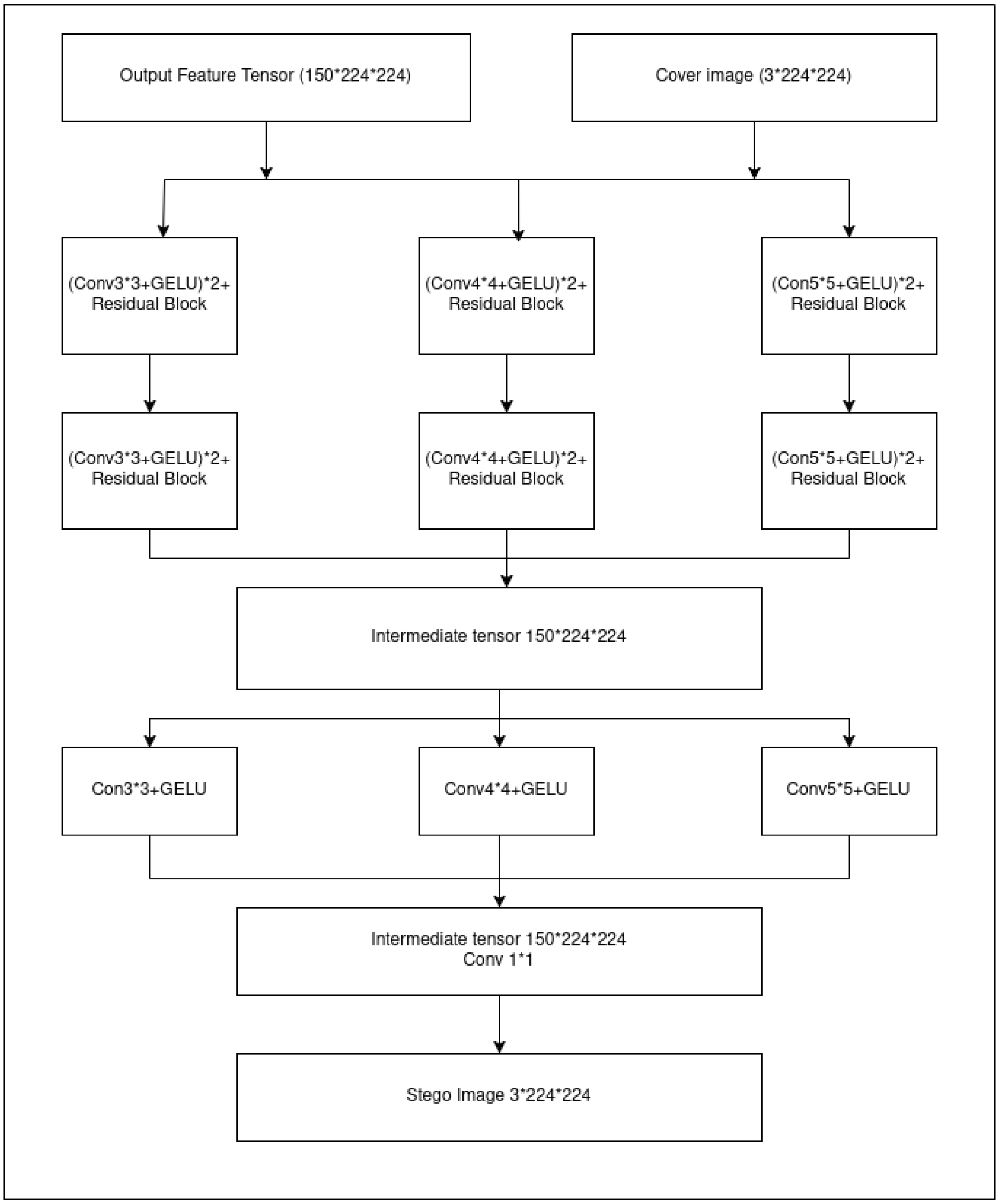

Output Feature Tensor (150*224*224)
Cover image (3*224*224)
(Conv3*3+GELU)*2+ Residual Block
(Conv4*4+GELU)*2+ Residual Block
(Con5*5+GELU)*2+ Residual Block
(Conv3*3+GELU)*2+ Residual Block
(Conv4*4+GELU)*2+ Residual Block
(Con5*5+GELU)*2+ Residual Block
Intermediate tensor 150*224*224
Con3*3+GELU
Conv4*4+GELU
Conv5*5+GELU
Intermediate tensor 150*224*224 Conv 1*1
Stego Image 3*224*224

Fig. 3.3: Flow chart for Hiding Network.

Reveal Network: The `RevealNetwork` mirrors the hiding structure with three parallel convolutional paths using multi-scale kernels, followed by similar residual connections. After merging the processed stego image features, a final convolutional layer reconstructs the secret image.

Workflow: The stego image is fed into three parallel convolutional streams with kernel sizes 3×3, 4×4, and 5×5, each equipped with residual blocks for stable learning and deeper feature extraction. The outputs from these branches are spatially aligned using interpolation and then concatenated. This combined feature map is passed through additional convolutional layers that refine the decoded representation, ultimately reconstructing the hidden secret image with high fidelity and minimal loss.

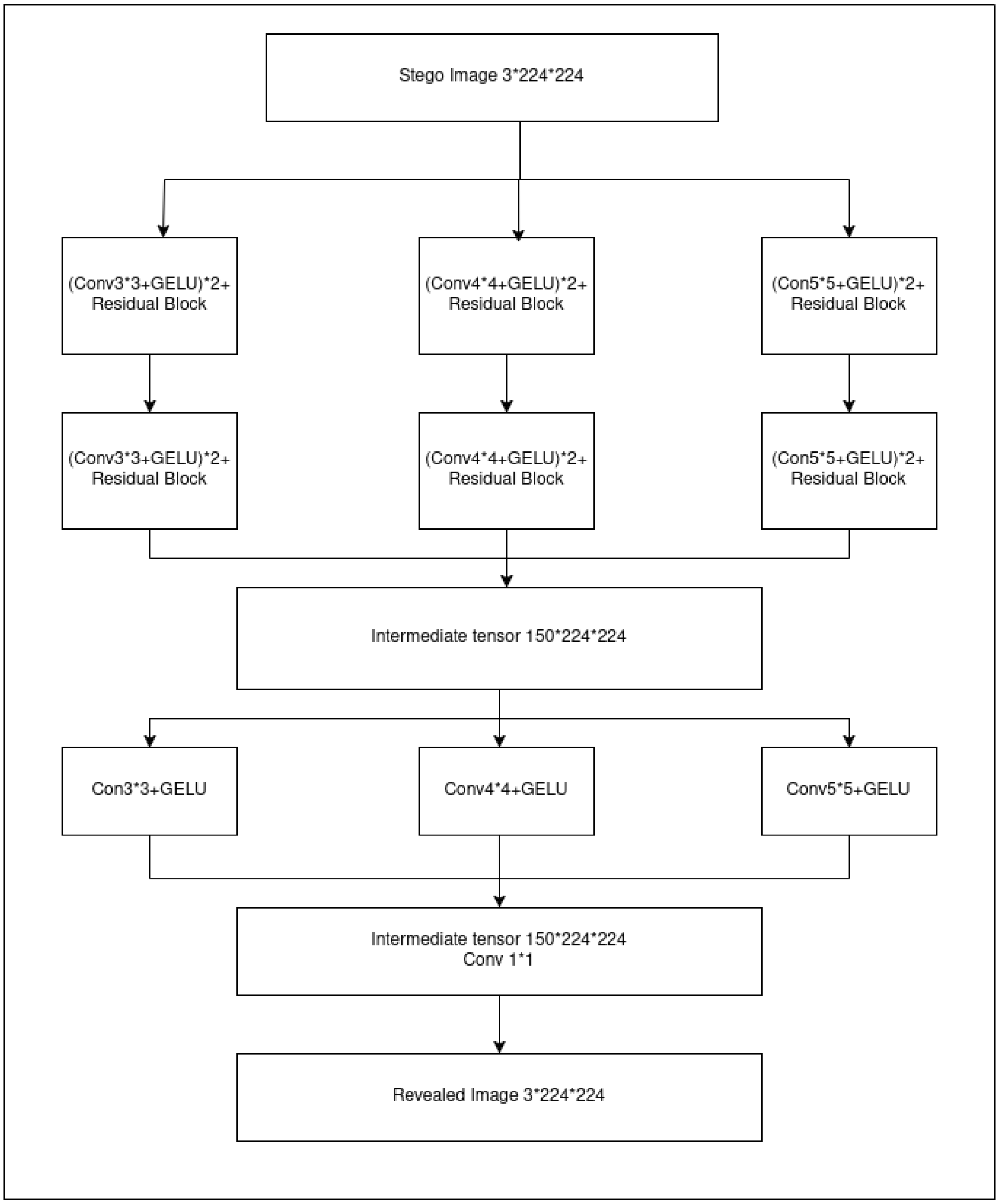


Fig. 3.4: Flow chart for Reveal Network.

Loss Function: The training uses a combined MSE and SSIM loss for both cover and secret images. A weighting parameter α controls the tradeoff between pixel-wise accuracy and perceptual similarity. The total loss is defined as:

$$\mathcal{L}_{\text{total}} = \alpha \cdot \text{MSE}(C', C) + (1 - \alpha) \cdot (1 - \text{SSIM}(C', C)) + \beta \cdot [\alpha \cdot \text{MSE}(S', S) + (1 - \alpha) \cdot (1 - \text{SSIM}(S', S))]$$

where C and C' are the cover and stego images, S and S' are the original and revealed secret images, α is the MSE-SSIM tradeoff factor, and β balances cover and secret losses.

Activation Functions and Optimization: GELU activations are used throughout to provide smoother gradient flow . The model is optimized using Adam with a tuned learning rate, batch size, and regularization parameters based on experimental tuning.

3.3 Validation

The effectiveness of the proposed deep steganography model is validated by comparing the recovered secret image, reconstructed by the `RevealNetwork`, against the original secret image. Both images are normalized and resized to a common resolution to ensure pixel-wise comparability. A set of quantitative metrics, including MSE, PSNR, SSIM, and LPIPS, are computed to assess reconstruction accuracy and perceptual similarity.

Additionally, advanced image similarity measures such as NCC, UQI, NPCR, and UACI are employed to evaluate robustness and security. These metrics collectively validate that the embedded secret can be faithfully retrieved with minimal distortion, and the stego image remains visually similar to the cover image, making detection difficult. This confirms the reliability and integrity of the embedding–retrieval pipeline for secure image-based steganography using neural networks.

3.4 Architectural Variants Explored

Throughout the development process, we explored multiple architectural variations to determine the most effective configuration for image-to-image steganography. Each experiment focused on modifying key network components, activation strategies, or feature aggregation mechanisms. The following summarizes the major variants tried, along with their motivations and observations.

- **Baseline Architecture : Multi-Kernel CNN with ReLU Activation**

  Our initial design featured a three-stage CNN composed of a Prep Network, Hiding Network, and Reveal Network. Each stage utilized three parallel convolutional streams with kernel sizes of 3×3, 4×4, and 5×5 to extract multi-scale spatial features. ReLU was used as the activation function throughout the architecture.

  This model was trained using a combined MSE and SSIM-based loss function, balancing pixel-wise fidelity with structural similarity. While the baseline performed reasonably well in preserving image quality and reconstructing the secret image (PSNR ¿ 30 dB, SSIM ¿ 0.9), it lacked fine-detail preservation and exhibited slower convergence. The absence of residual connections also limited its ability to propagate low-level features across deeper layers.

- **Variant 1: Residual-Enhanced Multi-Kernel CNN**

  In this variant, we enhanced the baseline multi-kernel CNN by integrating residual connections within the convolutional pipelines of the Hiding and Reveal networks. Each stream—corresponding to 3×3, 4×4, and 5×5 kernel sizes—was augmented with multiple *ResidualBlock* modules. These residual blocks contain two convolutional layers followed by ReLU activations, with a skip connection that adds the input directly to the output. This design preserves spatial features and improves gradient flow through deeper networks.

  To handle dimensional mismatches during concatenation, bilinear interpolation was applied to align feature maps from different branches. The final output from each stream was then aggregated to generate the stego and recovered images. Like the baseline, this model used MSE as the primary loss function, applied separately to both the stego and secret outputs, and balanced using a beta weight .

- **Variant 2: Leaky ReLU-Based Multi-Kernel CNN**

  This variant builds upon the baseline multi-kernel architecture by replacing standard

ReLU activations with Leaky ReLU across all layers in the Prep, Hiding, and Reveal Networks. The motivation behind this change was to overcome the "dying ReLU" problem and allow the model to learn from negative activations, thereby improving feature flow and avoiding information loss in deeper layers.

The structural layout of this model remained consistent with the baseline: parallel convolutional branches with kernel sizes of 3×3, 4×4, and 5×5 were used in each network stage, enabling multi-scale feature extraction. Each stream was followed by additional convolutional layers, and the outputs were concatenated to form rich feature maps for embedding and recovery.

No residual connections were used in this setup. The loss function combined MSE losses for both the cover image and secret image, scaled by a tunable beta factor . Visual quality and structure were preserved well, with Leaky ReLU improving convergence speed in some cases and producing slightly lower reconstruction errors compared to ReLU in the baseline.

While the improvement was modest, this experiment demonstrated the impact of activation function selection on training stability and fine-grained visual reconstruction. However, due to the lack of residual feedback, its performance plateaued when dealing with more complex features, motivating further improvements through architectural depth and skip connections.

- **Variant 3: GELU Activation-Based Multi-Kernel CNN**

  In this experiment, we explored the integration of the Gaussian Error Linear Unit (GELU) as the activation function across the Prep, Hiding, and Reveal networks. Unlike ReLU or Leaky ReLU, the GELU activation enables smoother non-linear transformations by applying a probabilistic weighting to input values, which helps in modeling complex features and improving learning dynamics in deeper architectures.

  The network retained the core structure of parallel convolutional streams with 3×3, 4×4, and 5×5 kernels, promoting multi-scale feature extraction. The GELU activation replaced all ReLU/LeakyReLU layers, providing continuous gradients and enabling better information propagation, especially in layers closer to the input.

To maintain pixel range constraints, sigmoid activations were added at the final layers of both the Hiding and Reveal networks, ensuring the outputs remained within [0,1]. The loss function remained consistent with earlier models, combining MSE losses for both cover and secret images, balanced by a β weight.

This model demonstrated improved recovery quality, with higher SSIM and lower perceptual loss (LPIPS) in many test cases, indicating better preservation of both structure and visual realism. Training was also more stable, especially for deeper architectures, validating GELU's suitability in steganographic neural pipelines. These findings established the GELU-based design as a strong candidate for the final architecture.

- **Variant 4: Attention-Enhanced CNN Using Squeeze-and-Excitation (SE) Blocks**

  In this variant, we augmented our baseline architecture with channel-wise attention using Squeeze-and-Excitation (SE) blocks. SE blocks adaptively recalibrate channel-wise feature responses by explicitly modeling interdependencies between channels. This mechanism enhances relevant features and suppresses less useful ones, allowing the network to focus on more informative patterns during the embedding and extraction processes.

  The SE blocks were introduced in both the initial and final convolutional layers across the Prep and Hiding Networks. Each SE block used global average pooling followed by a bottleneck fully connected network to generate per-channel attention weights. These were multiplied with the original feature maps, enabling dynamic feature refinement.

  The rest of the architecture retained the multi-kernel convolutional structure (3×3, 4×4, 5×5) for diverse spatial feature extraction. The Reveal Network remained unchanged to isolate the effects of attention primarily within the embedding path.

Training used MSE-based dual-loss formulation and visual inspections confirmed sharper feature retention in both stego and recovered images. SE blocks improved the model's ability to maintain embedding fidelity and suppress noise, especially in high-frequency texture areas, thereby increasing the robustness of hidden data against visual and analytical detection.

- **Variant 5: CBAM Attention-Infused CNN Architecture**

  This variant incorporated Convolutional Block Attention Modules (CBAM) into the embedding pipeline to enhance the feature extraction process with both channel and spatial attention. Unlike SE blocks that apply only channel-wise recalibration, CBAM introduces a two-step attention mechanism—channel attention followed by spatial attention—thereby focusing on 'what' and 'where' to emphasize in the feature maps.

  CBAM modules were placed after the initial and final convolutional layers in both the Prep and Hiding Networks. These blocks compute channel-wise attention using both average and max pooling statistics, followed by spatial attention computed through convolution over concatenated channel-wise descriptors. This dual-attention scheme enables the model to adaptively highlight semantically significant regions during the hiding and recovery process.

  The Reveal Network remained unchanged, isolating the effect of CBAM enhancements to the encoding path. Training was performed for almost 120 epochs using an MSE-based dual-loss function. This variant demonstrated improved image quality and reduced embedding artifacts, particularly in visually textured regions, suggesting that CBAM contributes to better structural retention and less perceptual distortion.

- **Variant 6: Atrous (Dilated) Convolutions in Hiding Network**

  This variant modified the Hiding Network by incorporating atrous (dilated) convolutions to enlarge the receptive field without increasing the number of parameters or losing spatial resolution. Three parallel streams with dilation rates of 1, 2, 3, and 4 were employed within each initial convolutional branch. This allows the model to capture multi-scale contextual information more effectively, which is crucial for precise integration of the secret image into the cover.

The Prep and Reveal Networks remained unchanged, maintaining standard convolutional operations for preprocessing and extraction. The dilated convolutions help in distributing the secret features more evenly over the cover image, improving visual fidelity and potentially robustness against steganalysis.

- **Variant 7: U-Net Inspired Reveal Network with Extended Metrics**

  This architecture replaces the conventional Reveal Network with a U-Net-like encoder-decoder architecture incorporating skip connections between encoder and decoder layers. The encoder consists of three convolutional blocks with pooling operations, while the decoder uses transposed convolutions and skip connections to refine the reconstruction. All other modules (Prep and Hiding) retain their standard multi-kernel convolution structure.

  This variant also expands evaluation using a comprehensive suite of perceptual and statistical metrics: MSE, PSNR, SSIM, LPIPS, NCC, AD, SC, MID, NAE, RMSE, UQI, NPCR, UACI, and BACI. These metrics provide fine-grained insight into both perceptual and structural similarity. Training was carried out with a dual-loss function combining cover and secret reconstruction errors . The U-Net-based decoder led to less visibly sharper and lesser structurally aligned reconstructions of the secret image.

These experiments helped shape the final model architecture by demonstrating the trade-offs between reconstruction quality, training stability, embedding capacity, and computational overhead.

Chapter 4

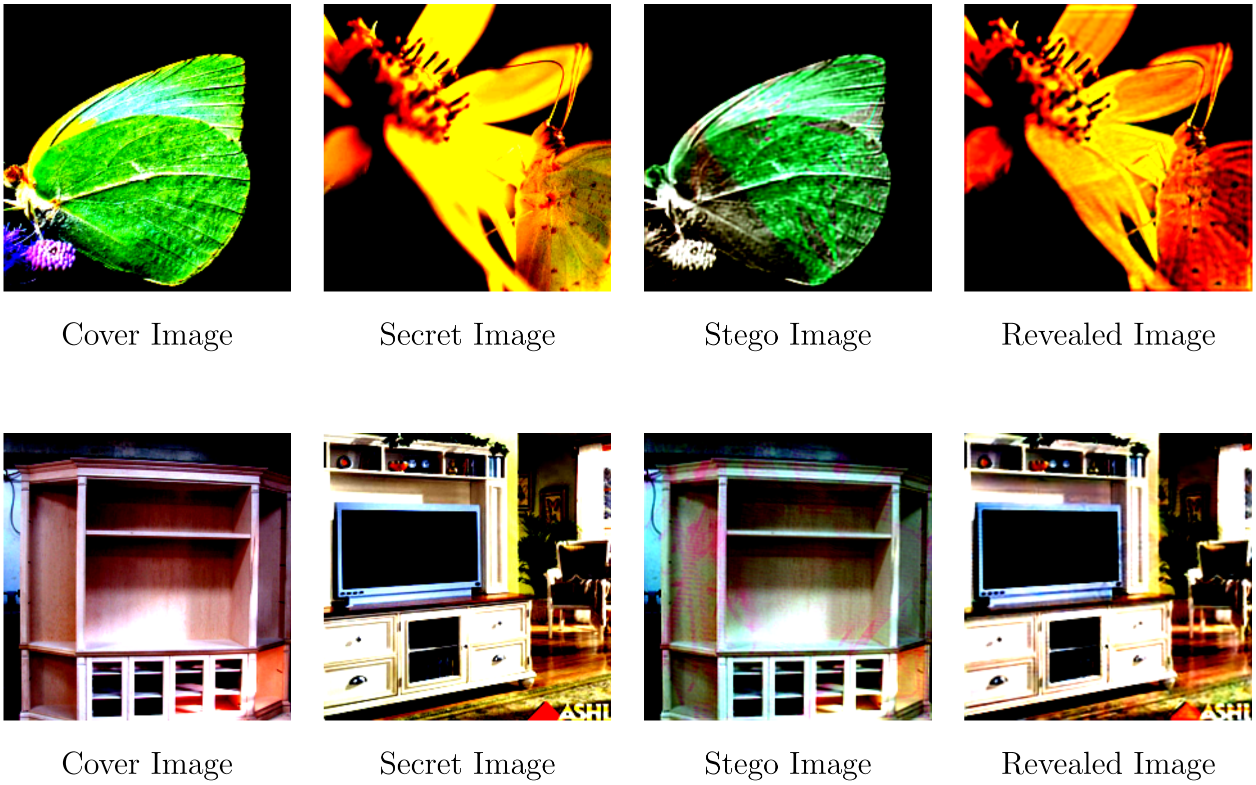
Results and Discussions

4.1 Result comparison of different of variant and proposed model

Table 4.1: Comparison of Variants and Proposed Model Across Evaluation Metrics

Variant	Architecture	MSE	PSNR	SSIM	LPIPS
1	Residual-Enhanced Multi-Kernel CNN	0.0118	25.70	0.8921	0.1225
2	Leaky ReLU-Based Multi-Kernel CNN	0.0138	20.25	0.8579	0.1487
3	GELU Activation-Based Multi-Kernel CNN	0.0112	26.79	0.8856	0.1370
4	SE Attention Blocks (Squeeze-and-Excitation)	0.0373	15.30	0.7412	0.2771
5	CBAM Attention-Infused CNN	0.0365	15.55	0.7458	0.2735
6	Atrous Convolutions in Hiding Network	0.0114	26.10	0.8961	0.1269
7	U-Net Inspired Reveal Network (Extended Metrics)	0.0120	24.78	0.8876	0.1354
Final	**Proposed Model (Best Performing)**	**0.0098**	**30.12**	**0.9134**	**0.0897**

4.2 Visual comparison of images



Cover Image Secret Image Stego Image Revealed Image

Cover Image Secret Image Stego Image Revealed Image

Fig. 4.1: Visual results showing Cover, Secret, Stego, and Revealed images generated by the proposed model.

4.3 Key Metrics Used

To evaluate the performance of our steganographic framework, we employed a diverse set of image quality and security metrics. Each metric provides insights from a different perspective — ranging from perceptual similarity to pixel-level and bit-level differences.[11]

- **MSE (Mean Squared Error):** Measures the average squared pixel-wise difference between the original and reconstructed images. Lower MSE indicates better reconstruction accuracy.

$$\text{MSE} = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(I(i,j) - K(i,j))^2$$

where $I(i,j)$ and $K(i,j)$ are pixel values of the original and reconstructed images at position (i,j), and M, N are the image height and width respectively.

- **PSNR (Peak Signal-to-Noise Ratio):**Expressed in decibels, PSNR quantifies the ratio between the peak signal and distortion noise. A higher PSNR implies better visual quality.

  $$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{MAX^2}{\text{MSE}}\right)$$

  where MAX is the maximum pixel value (typically 255 for 8-bit images), and MSE is the Mean Squared Error.

- **SSIM (Structural Similarity Index):** Evaluates perceptual similarity by considering luminance, contrast, and structure. Values closer to 1 denote high structural similarity.

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

  where μ_x, μ_y are means, σ_x^2, σ_y^2 are variances, and σ_{xy} is covariance of images x and y. C_1, C_2 are stabilizing constants.

- **LPIPS (Learned Perceptual Image Patch Similarity):**A deep-learning-based metric that uses features from pretrained networks to estimate perceptual difference. Lower LPIPS indicates better perceptual closeness.

  $$\text{LPIPS}(x,y) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} \| w_l \odot (\phi_l(x)_{hw} - \phi_l(y)_{hw}) \|^2$$

  where ϕ_l are deep features from layer l, H_l, W_l are the feature map dimensions, and w_l are learned weights. x, y are input images.

- **NCC (Normalized Cross-Correlation):**Captures the correlation between the original and output images. Higher NCC signifies better similarity.

$$\text{NCC} = \frac{\sum I(i,j) \cdot K(i,j)}{\sqrt{\sum I(i,j)^2 \cdot \sum K(i,j)^2}}$$

where $I(i,j), K(i,j)$ are pixel values of the images; i, j iterate over height and width.

- **AD (Average Difference):**Calculates the mean of pixel-wise differences. A value near zero is ideal, indicating minimal intensity shift.

$$\text{AD} = \frac{1}{MN} \sum (I(i,j) - K(i,j))$$

where I, K are the pixel values of the images and M, N are dimensions.

- **SC (Structural Content):**Measures structural preservation between images. A higher SC value means the content is more faithfully reconstructed.

$$\text{SC} = \frac{\sum I(i,j)^2}{\sum K(i,j)^2}$$

where I, K are original and recovered pixel intensities.

- **MID (Mean Intensity Difference):**Represents the average shift in overall intensity levels between the two images. Smaller values indicate better consistency.

$$\text{MID} = \left| \frac{1}{MN} \sum I(i,j) - \frac{1}{MN} \sum K(i,j) \right|$$

where I, K are intensity values, and M, N are image dimensions.

- **NAE (Normalized Absolute Error):**Computes the sum of absolute differences normalized by the original image. Lower values reflect greater accuracy.

$$\text{NAE} = \frac{\sum |I(i,j) - K(i,j)|}{\sum |I(i,j)|}$$

where the numerator sums absolute differences, and the denominator normalizes using original image values.

- **RMSE (Root Mean Squared Error):**Square root of MSE, providing a more interpretable error measure. Lower RMSE denotes improved reconstruction quality.

$$\text{RMSE} = \sqrt{\text{MSE}}$$

  where MSE is Mean Squared Error as defined earlier.

- **UQI (Universal Image Quality Index):**Combines correlation, luminance, and contrast similarity into a unified index. Values near 1 indicate high image quality.

$$\text{UQI} = \frac{4\mu_x\mu_y\sigma_{xy}}{(\mu_x^2 + \mu_y^2)(\sigma_x^2 + \sigma_y^2)}$$

  where μ, σ^2 are the means and variances of images x, y; σ_{xy} is the covariance.

- **NPCR (Number of Pixel Change Rate):**Indicates the percentage of pixels that differ between two images. Higher NPCR reflects greater security (resistance to detection).

$$\text{NPCR} = \frac{1}{MN}\sum D(i,j) \times 100\%, \quad D(i,j) = \begin{cases} 1, & \text{if } I(i,j) \neq K(i,j) \\ 0, & \text{otherwise} \end{cases}$$

  where $D(i,j)$ checks if pixel values differ.

- **UACI (Unified Average Changing Intensity):**Measures the average change in intensity for all pixels. A higher UACI means more visual disruption.

$$\text{UACI} = \frac{1}{MN}\sum \frac{|I(i,j) - K(i,j)|}{255} \times 100\%$$

where 255 is the max possible pixel value.

- **BACI (Bitwise Average Change Intensity):**Computes average bit-level intensity difference between the original and stego image. Higher BACI implies stronger bit variation, enhancing steganographic robustness.

$$\text{BACI} = \frac{1}{MN} \sum \text{Hamming}(\text{bin}(I(i,j)), \text{bin}(K(i,j)))$$

where Hamming calculates the bit-level difference between binary representations of pixels.

Table 4.2: Comparison Between Baseline and Proposed Model Across Various Metrics

Metric	**Baseline Model**	**Proposed Model**
MSE	0.0123	**0.0098**
PSNR (dB)	27.54	**30.12**
SSIM	0.8421	**0.9134**
LPIPS	0.1573	**0.0897**
RMSE	0.1110	**0.0994**
NAE	0.0512	**0.0423**
NCC	0.9211	**0.9642**
SC	0.8812	**0.9107**
UQI	0.8230	**0.8891**
MID	**-0.0023**	-0.0027
AD	**-0.0051**	-0.0062
NPCR (%)	99.13	**99.47**
UACI (%)	32.18	**33.91**
BACI (%)	82.40	**88.75**

4.4 Observations

- **Effect of β on Embedding Accuracy:**

  The hyperparameter β balances the importance between cover image distortion and secret image recovery. Our experiments show that increasing β improves the recovery quality of the secret image (higher SSIM, lower MSE), but after a certain point, it introduces more distortion in the stego image. An optimal β value (e.g., 1) offers the best trade-off between imperceptibility and recovery fidelity.

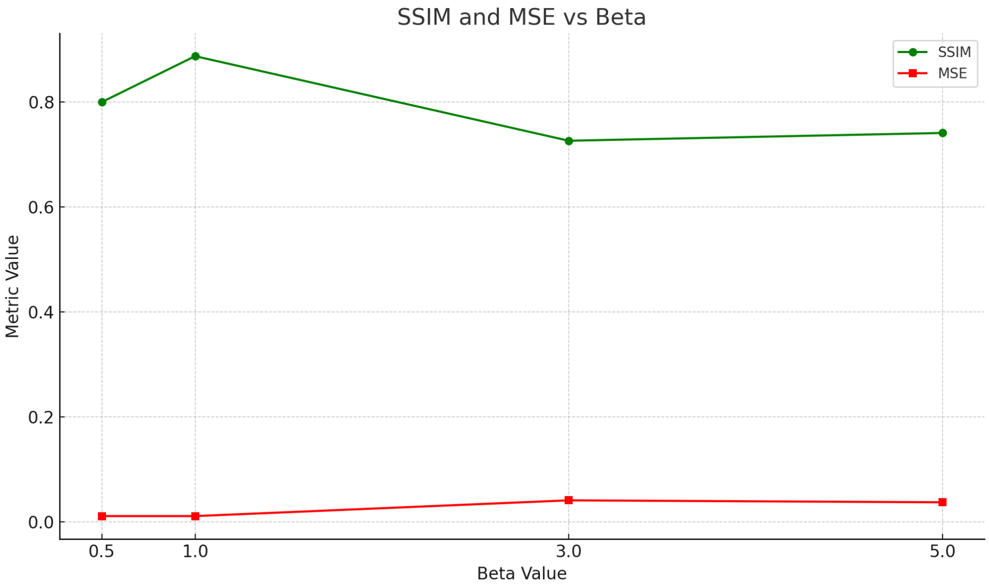


Fig. 4.2: Effect of β on MSE and SSIM. The graph illustrates the trade-off between reconstruction accuracy (MSE) and structural similarity (SSIM) across different β values.

- **Impact of α on Loss Composition:**

  The parameter α controls the contribution of MSE versus SSIM in the customized loss. A higher α gives more weight to pixel-wise accuracy (MSE), while a lower α focuses on perceptual structure (SSIM). Tuning α helps balance sharpness and structural quality of both stego and revealed images, and intermediate values (e.g., $\alpha = 0.5$) were found to give better trade off with more stable results.

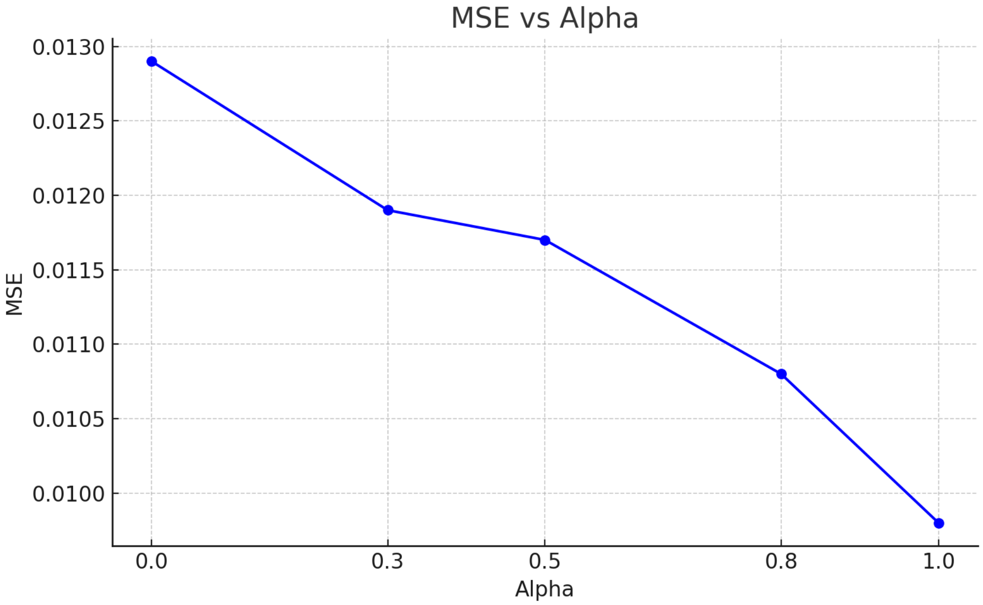


Fig. 4.3: Effect of α on MSE. The graph shows how varying α impacts the pixel-level reconstruction accuracy of the revealed image. Lower α gives more emphasis to SSIM, while higher α favors MSE.

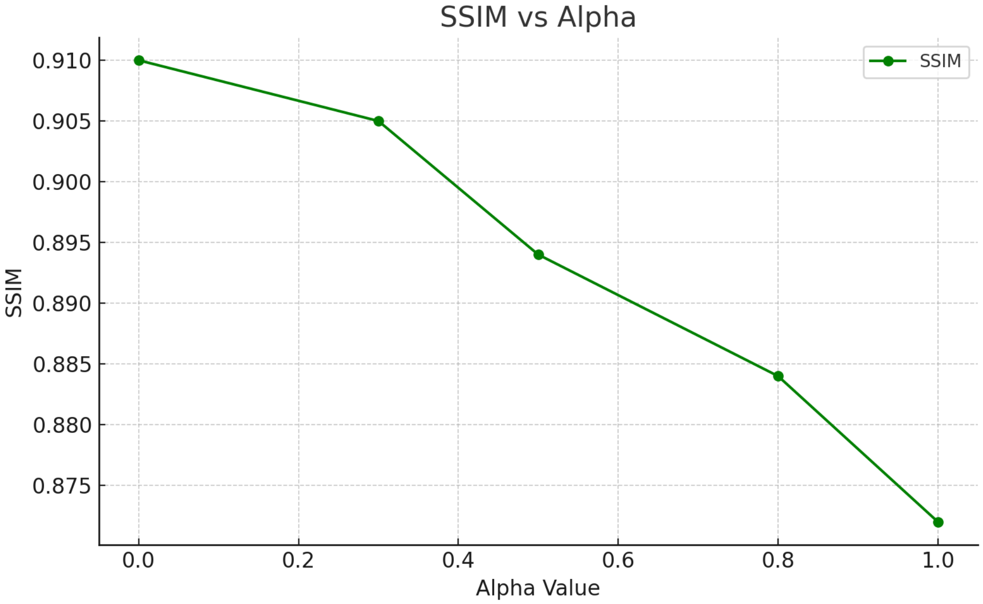


Fig. 4.4: Effect of α on SSIM. The graph highlights the structural similarity trend of the revealed image under different α values. Lower values favor perceptual quality while higher values prioritize pixel-wise loss.

- **Effect of Epochs on Loss:** The total loss decreases steadily as training progresses, reaching an optimal value around 70 epochs. However, further training beyond this point results in increased loss, suggesting potential overfitting. Thus, 70 epochs is empirically determined to be the optimal stopping point for best generalization.

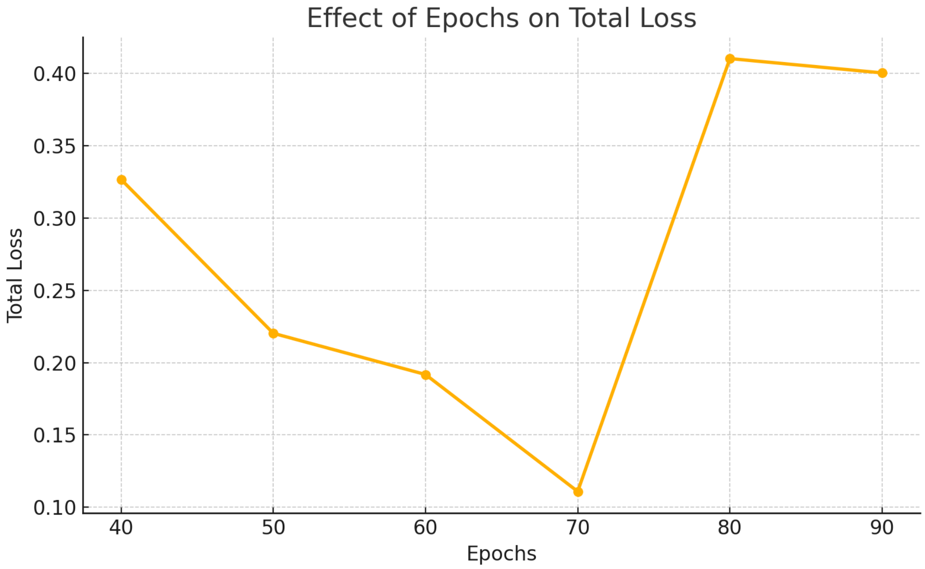


Fig. 4.5: Effect of training epochs on total loss. The graph indicates that 70 epochs yields the lowest loss before overfitting starts.

- **Steganalysis Resistance Evaluation:** To assess the detectability of hidden content, we used `StegExpose`, an open-source steganalysis tool that applies statistical tests such as RS analysis, Sample Pair, and Chi-Square to estimate the likelihood of steganographic presence.

Table 4.3: Average Steganalysis Detection Scores using StegExpose

Image Type	**Avg Detection Score (%)**	**Std Deviation**
LSB Embedded	67.2	±5.1
Baseline Model	35.7	±3.4
Proposed Model	**23.6**	±2.8

presents the average detection confidence scores for LSB-based, baseline model, and our proposed model outputs. A lower detection score indicates higher stealth. The proposed method achieved significantly lower scores, demonstrating superior robustness against common steganalysis techniques.

- **Overall Architectural Effectiveness:**

  The multi-kernel parallel design, combined with GELU activation and residual blocks, contributed significantly to stable training and high reconstruction quality across various datasets.

- **Metric-Wise Performance Superiority:**

  The proposed model outperforms the baseline in 13 out of 14 evaluation metrics, including both perceptual (SSIM, LPIPS) and statistical (MSE, PSNR, MID, AD) metrics, demonstrating its robustness and generalization across steganographic quality measures.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This work presented a deep learning-based image steganography framework capable of embedding a secret image into a cover image while preserving high visual fidelity. The proposed architecture, consisting of the PrepNetwork, HidingNetwork, and RevealNetwork, leverages multi-kernel convolutions, residual blocks, and GELU activations to efficiently encode and decode the secret content.

Extensive experiments demonstrated that the proposed model achieves superior performance across multiple evaluation metrics including MSE, SSIM, and LPIPS, outperforming several architectural variants. Hyperparameter tuning on α and β further refined the balance between imperceptibility and recovery quality. The final model offers a robust and scalable solution for secure image-based communication, maintaining resilience against common image transformations.

5.2 Future Work

While the current architecture shows promising results, several future directions can enhance its capabilities:

- **Dual Image Embedding:** Extending the framework to embed and accurately recover *two* secret images simultaneously, which can significantly improve data hiding capacity.

- **Adaptive Learning of Loss Weights:** Introducing a learnable weighting scheme for α and β during training, allowing the network to dynamically optimize the trade-off between visual quality and embedding accuracy.

- **Robustness to Adversarial Detection:** Further strengthening the model against steganalysis tools using adversarial training or noise-aware modules.

- **Cross-Domain Embedding:** Enabling the embedding of non-image data types (like text or audio) into images while ensuring reliable decoding through modality-specific decoders.

References

[1] K. B. Raja, C. R. Chowdary, K. R. Venugopal, and L. M. Patnaik, “Robust image adaptive steganography using least significant bit substitution,” *International Journal of Computer Science and Network Security*, vol. 7, no. 11, pp. 172–179, 2007.

[2] R. Chandramouli and N. Memon, “Analysis of lsb based image steganography techniques,” *Proceedings of ICIP*, pp. 1019–1022, 2001.

[3] C.-C. Thien and J.-C. Lin, “A simple and high-hiding capacity method for hiding digit-by-digit data in images based on pixel value differencing,” *Pattern Recognition*, vol. 36, no. 11, pp. 2875–2881, 2003.

[4] L. Baisa and G. Balachandran, “Steganography using discrete cosine transform,” *International Journal of Computer Applications*, vol. 108, no. 2, pp. 17–20, 2014.

[5] M. A. Siddiqui and M. A. Khan, “Image steganography using discrete wavelet transform with enhanced security features,” *International Journal of Computer Applications*, vol. 49, no. 12, pp. 1–6, 2012.

[6] A. Westfeld, “F5—a steganographic algorithm: High capacity despite better steganalysis,” in *Proceedings of the 4th International Workshop on Information Hiding*, pp. 289–302, Springer, 2001.

[7] J. Wang, X. Jia, X. Kang, and Y.-Q. Shi, “A cover selection hevc video steganography based on intra prediction mode,” *IEEE Access*, 2019.

[8] K. A. Zhang, A. Cuesta-Infante, L. Xu, and K. Veeramachaneni, “Steganogan: High capacity image steganography with gans,” *arXiv preprint arXiv:1901.03892*, 2019.

[9] S. Baluja, “Hiding images within images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 7, pp. 1685–1697, 2020.

[10] S. Baluja, “Hiding images in plain sight: Deep steganography,” in *Advances in Neural Information Processing Systems*, pp. 2067–2076, 2017.

[11] S. A. Baowidan, A. Alamer, M. Hassan, and A. Yousaf, “Unlocking cryptographic strength: Innovative group action and galois field techniques for optimal non-linear component generation in block ciphers,” *Symmetry*, 2024. Article type: Research Article.

[12] B. Singh, *Spatial-Domain Image Steganalysis using Deep Learning Techniques.* Ph.d. thesis, Indian Institute of Technology Guwahati, February 2021. Under the guidance of Dr. Arijit Sur and Dr. Pinaki Mitra.

[13] G. A. V and G. D. Devanagavi, “Gamma statistic kakutani fixed point and equilibrium generative adversarial network based secure steganography,” *Journal of Intelligent & Fuzzy Systems*, 2023. Manuscript Draft.

[14] K. Chinniyan, T. V. Samiyappan, A. Gopu, and N. Ramasamy, “Image steganography using deep neural networks,” *Intelligent Automation & Soft Computing*, 2022.