Acknowledgement

We would like to thank our faculty and management for their professional guidance towards the completion of the project work. We take this opportunity to thank Dr. Ashok Shettar, Vice-Chancellor, Dr. B.S Anami, Registrar, and Dr. P.G Tewari, Dean Academics, KLE Technological University, Hubballi, for their vision and support.

We also take this opportunity to thank Dr. Meena S. M, Professor and Head, SoCSE for having provided us direction and facilitated for enhancement of skills and academic growth.

We thank our guide Dr.G.S.Hanchinamani, SoCSE for the constant guidance during interaction and reviews.

We extend our acknowledgement to the reviewers for critical suggestions and inputs. We also thank Project Co-ordinator Mr. Uday N.Kulkarni and r. Guruprasad Konnuramath for their support during the course of completion.

We express gratitude to our beloved parents for constant encouragement and support.

Akash Kumar - 01fe20bcs300
Apoorv Bagal - 01fe20bcs065
Abhishek Kumar - 01fe20bcs059
Yejas Menon - 01fe20bcs048

ABSTRACT

The practise of steganography, which involves hiding information in seemingly innocent media, has becoming more important in the digital age. A study on steganography utilising convolutional neural networks (CNNs) is presented in this work. The CNN-based method effectively conceals information within images by utilising the potent image analysis and feature extraction capabilities of CNNs.In order to implant the secret information, the suggested method relies on changing the least significant bits (LSBs) of particular pixels throughout the encoding procedure. The CNN-extracted local picture features serve as a guide for these adjustments, assuring imperceptibility while preserving the cover image's aesthetic quality.A trained CNN model is used for decoding in order to extract the concealed information from the stego image. The CNN can accurately recover the encoded information thanks to its capacity to learn complex visual patterns and features. The efficiency of the CNN-based steganographic method in terms of imperceptibility, capacity, and resilience against different steganalysis techniques is demonstrated by experimental results.The CNN-based steganographic method is appropriate for a variety of applications that require concealed information within images, while maintaining visual fidelity and ensuring robust security, thanks to the combination of deep learning and image analysis techniques.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Utilising non-secret text or data to cloak messages or information is a method known as steganography. The term is a combination of the Greek words "steganos," which means covered or veiled, and "graphia," which means writing. Steganography's main objective is to conceal a message's existence in a way that prevents unauthorised parties or unintended receivers from noticing it.

Steganography seeks to secure confidential information by enclosing sensitive data within seemingly innocent carriers, such as pictures, audio files, video clips, or even text pages. Steganography concentrates on hiding the existence of a communication, as opposed to cryptography, which is concerned with encrypting the contents of a message.

Steganographic hiding can be achieved using a variety of techniques. One such method is changing the least important parts of a digital asset, like an image or audio sample, in order to encode the secret information. The carrier file seems unmodified since these modifications are frequently invisible to the human eye, successfully hiding the contained message.

Steganography may be used for both good and bad intentions. It can be utilised for clandestine communication, digital watermarking, or copyright protection in appropriate circumstances. For instance, to securely send sensitive material, journalists working under oppressive governments may use steganography.

It is now simpler to accomplish and more difficult to find because to new technology, making it a useful method for people to hide sensitive information from others who want to see it.

The method of discovering and analyzing concealed data has been dubbed "steganalysis" in order to combat the abuse of steganography. To find suspicious trends or departures from typical file features, these techniques include statistical analysis, anomaly detection, and machine learning algorithms.

Steganography techniques are constantly changing as a result of technological advancements, necessitating continuing study and the creation of reliable detection systems. Un-

derstanding steganography and its goals can help people and organisations see the possible threats and take the necessary security precautions to protect their data.

1.1 Motivation

- The likelihood of harmful threats, eavesdropping, and other subversive operations has increased due to the accessibility of several freely available instruments.
- These tools are able to violate the confidentiality, integrity, and security of the sent data.
- Steganography makes it difficult for unauthorised people to obtain information by concealing it in a picture.
- The ability of traditional steganography to conceal information is limited. The size of the message to be buried determines how obvious the changes will be.
- The resilience of steganography approaches can be increased because CNNs are known for their capacity to extract robust features from complex data.
- The amount of hidden information that can be encoded in stego pictures can be increased thanks to CNNs' excellent capacity for learning complex patterns in data.
- In terms of processing time and storage needs, CNN-based steganography algorithms may be more effective, making them more applicable for use in practical settings.
- Compared to conventional steganography techniques, CNN-based steganography approaches are more adaptable to various sorts of data, including photos, audio, and video.
- Steganalysis tools are capable of detecting conventional steganography procedures.

1.2 Literature Survey

This paper[1] gives a general review of steganography and discusses its value in the current digital era.Convolutional neural networks (CNNs) are suggested as a steganography technique to conceal messages in digital photographs. By utilising deep learning to raise the hiding capacity and decrease the detectability of the hidden messages, this method seeks to increase the security and robustness of steganography systems.The two steps of the suggested approach are preparation and embedding. The cover picture is processed using a CNN during the preparation stage to produce a feature map, which is then utilised to get the cover image ready for embedding. The secret message is first translated to binary in the embedding process, and then it is added to the prepared cover image by changing the pixels' least significant bits.The authors experimented on several datasets to gauge the efficiency of the suggested method, then compared the findings with those obtained using other steganography techniques. The results of the experiments demonstrate that the suggested strategy works better than the current methods in terms of hiding ability and detection rate. The suggested method is also demonstrated to be resilient to a variety of image processing procedures, including noise and compression.The research offers a fresh method for enhancing the security and resilience of steganography approaches, and it also illustrates the possibilities of using CNNs for steganography.

The purpose of the paper [2] is to evaluate the performance of the suggested approach using experimental data.The importance of steganalysis in the context of digital forensics and the need for better methods of identifying steganographic content are covered in the opening section of the study. The authors then put forth a fresh strategy that boosts identification accuracy by combining detailed model features with convolutional neural networks (CNNs).A genetic algorithm is employed in the feature selection step to choose the most pertinent features from the set of retrieved features. This increases the effectiveness of the classification step and lowers the dimensionality of the feature space.A CNN is trained to categorise the image as either stego or cover at the end of the classification process. The CNN uses the chosen features as input and produces a likelihood score for each class. Additionally, the authors suggest a brand-new loss function that is intended to more effectively address the class imbalance issue that can occur in steganalysis.The three primary steps of the suggested method are feature extraction, feature selection, and classification. A set of features are extracted from the image using a rich model in the feature extraction step. This contains deep learning features that were taken from a pre-trained CNN as well as more conventional hand-crafted

features like histogram-based features and co-occurrence matrix features.The experimental findings reported in the research show that, in terms of detection accuracy on a common benchmark dataset, the suggested strategy surpasses a number of cutting-edge steganalysis methods. The authors come to the conclusion that the use of CNN's in conjunction with rich model features has the potential to increase the efficacy of steganalysis methods.

In this paper [3], the authors propose a simple and efficient approach that utilizes the LSBs of image pixels for data hiding.The paper provides a detailed explanation of the embedding and extraction procedures. By using the bits from the secret message, LSB replacement replaces the least significant bit of the pixel values. This method is well-liked because it allows you to conceal a lot of info in an image without significantly lowering the image quality.The "sequential least significant bit" (SLSB) approach, a special variation of LSB substitution, is the main topic of this study. The secret message is broken up into bits and successively inserted into the least important bits of the image pixels using this manner, starting in the top-left corner and progressing to the right. The efficiency of this technique is examined in the study using a variety of statistical tests to find any concealed information in the image.The results of the studies demonstrate that the SLSB method can survive various steganalysis techniques while being effective in concealing data in photos. The technique is not foolproof, though, and can be discovered by advanced steganalysis techniques that scan an image for steganography-specific patterns. In order to make the SLSB approach more resistant to detection, the study discusses some potential improvements that could be made.

In this paper [4] , they have proposed a new embedding cost learning framework called SPAR-RL(Steganography Pixel-wise Actions and Rewards with Reinforcement Learning).The article suggests a novel method for enhancing the cost function for image steganography. Traditional steganography techniques rely on manually created cost functions, which can be unreliable and may not adequately capture the intricate connections between the cover and stego images.The suggested framework uses steganography and deep reinforcement learning (DRL) to learn the best cost function for creating high-quality stego pictures. The feedback it receives from the environment helps the DRL agent decide on the best steganographic embedding techniques. The environment is made up of a steganalysis system that rates how well the agent's stego pictures are produced.On the BOSSbase and BOWS-2 datasets, the sug-

gested method is assessed and contrasted with cutting-edge steganography techniques. The outcomes demonstrate that the suggested strategy exceeds current approaches in terms of embedding rate and detection precision. Additionally, it is demonstrated that the method can produce excellent stego pictures and is resistant to attacks.Overall, the paper offers a promising method for leveraging deep reinforcement learning to optimise the cost function for steganography. The method may enhance the reliability and security of steganographic systems and enhance the security of sensitive information stored in images.

The paper [5] works by examining the adjacent pixels in an image and calculating the difference between their values. The secret data is then embedded by modifying these pixel differences based on the binary representation of the hidden information. The modified pixel differences are typically imperceptible to the human eye, ensuring that the stego image appears visually similar to the original image.Paper discusses the effectiveness and security of the PVD method through experimental results and analysis. It compares the proposed technique with other existing steganographic methods, demonstrating its advantages in terms of imperceptibility and resistance against attacks. The authors also discuss the limitations and potential areas for further research in the field of steganography.

The research paper [6] introduces an innovative approach to strengthen the security of image steganography. By combining the capabilities of neural networks and visual cryptography, the proposed method aims to create a more secure and robust system.The proposed method has demonstrated superior security compared to conventional image steganography techniques. It exhibits increased resilience against steganalysis attacks, enhancing the system's overall robustness. Although the method is still in the development phase, it holds significant potential to revolutionize the field of image steganography by providing enhanced security measures.Some of the key advantages of the proposed method include heightened security, achieved through the combined utilization of neural networks and visual cryptography. The intricate nature of the approach makes it exceedingly difficult for an attacker to recover the secret data. Additionally, the method showcases robustness against steganalysis attacks, surpassing traditional image steganography methods. As the method continues to undergo development, it presents opportunities for future advancements and improvements, promising a significant impact on the field of image steganography.

The authors of the paper [7] highlight image-to-image steganography, which involves hiding one image within another, is a well-studied topic, the challenge of concealing two secret images in a single carrier image remains significant. Moreover, the practical application of image steganography based on deep learning techniques is relatively limited. To address these challenges, the authors propose the SteganoCNN model, which consists of two main modules: an encoding network and a decoding network. The decoding network comprises two extraction networks. The entire network is trained end-to-end, with the encoding network automatically embedding the secret images into the carrier image, while the decoding network is responsible for reconstructing the two different secret images. The experimental results demonstrate that the proposed steganography scheme achieves a maximum image payload capacity of 47.92 bits per pixel. Additionally, the scheme effectively evades detection by steganalysis tools while preserving the quality of the stego-image. The authors also highlight that SteganoCNN exhibits strong generalization capabilities, allowing the steganography of various data types, such as remote sensing images and aerial images.

The paper [8] focuses on the application of deep learning techniques for analyzing and detecting steganographic content in images. It proposes a novel approach that utilizes deep neural networks to identify hidden information and uncover potential steganographic techniques used.The authors highlight the increasing sophistication of steganographic methods and the need for advanced detection techniques. They introduce a deep learning framework specifically designed for steganalysis, which involves training a convolutional neural network (CNN) on a large dataset of both cover and stego images. The CNN is trained to learn discriminative features that can distinguish between normal images and those containing hidden data.The paper describes the methodology of building the deep learning model, including the architecture of the CNN and the training process. It emphasizes the importance of a diverse and representative dataset for achieving accurate detection results.The experimental evaluation section presents the results of the proposed deep learning approach. The model is evaluated on different benchmark datasets, and its performance is compared with other traditional steganalysis methods. The authors report promising results, demonstrating the effectiveness of deep learning in detecting steganographic content with higher accuracy and efficiency.

The paper concludes by discussing the implications and future directions of utilizing deep learning for steganalysis. It emphasizes the potential of deep neural networks in detecting advanced steganographic techniques and the need for ongoing research and development in this area. The authors also highlight the importance of continuously updating the detection models to adapt to evolving steganography methods.

The Paper [9] addresses the problem of transmitting biomedical images securely over a public network. The paper highlights that when medical records are sent in plain form, they face issues related to confidentiality and integrity. To overcome this problem, the authors propose a scheme where medical records are hidden within a cover image to ensure privacy during transmission. In the proposed scheme, the cover image is divided into three color planes: R (red), G (green), and B (blue). The co-occurrence matrix is computed for each plane by dividing it into 16x16 pixel blocks, and an embedding map is generated from it. The biomedical image is also divided into 8x8 pixel blocks and hidden within the corresponding blocks of the cover image using the embedding map. Both the cover image blocks and secret image blocks are transformed by Recursive Integer Wavelet Transform (RIWT). The R matrix of the QR decomposition of the cover image and secret image blocks is used for embedding. Finally, the three planes are merged into an RGB image, resulting in a stego image.The performance of the proposed scheme is evaluated based on imperceptibility, robustness, and security. The stego images exhibit high imperceptibility, as indicated by peak signal-to-noise ratio (PSNR) values exceeding 50. The robustness of the scheme is demonstrated by the average normalized correlation coefficient (NCC) values between the original secret image and the attacked secret image, which is 0.94 on average. Regarding security, the stego image remains undetectable even if it contains a hidden secret.Based on the experimental results, the authors conclude that their proposed scheme outperforms existing approaches in terms of imperceptibility, robustness, and security. The paper contributes to the field of image steganography, specifically in the context of transmitting biomedical images securely in cloud environments.

The authors of this paper [10] introduces a new method for steganalysis, focusing on detecting hidden information in digital media, particularly images. The authors propose a deep learning approach that utilizes convolutional neural networks (CNNs) to learn expressive features for efficient steganalysis. The research aims to address the limitations of traditional steganalysis methods that rely on handcrafted features and struggle with low embedding rate images and multi-class steganography scenarios. In contrast, deep learning models can automatically learn discriminative features from the data, leading to more accurate and robust steganalysis.The authors present their CNN architecture, which incorporates multiple convolutional and pooling layers to extract hierarchical features from input images. They also discuss the training process, including data preprocessing and augmentation techniques to improve the model's generalization ability.To evaluate the proposed method, the authors conduct experiments on benchmark datasets with various steganography algorithms. The results demonstrate that their deep learning approach outperforms traditional methods, achieving

higher detection rates and lower false positive rates. The authors also perform ablation studies to analyze the contribution of different components in their model.The paper concludes by discussing the implications of the research in steganalysis. It emphasizes the potential for further advancements in deep learning approaches, such as exploring different network architectures and training strategies. The authors suggest that incorporating larger and more diverse datasets could lead to even more effective steganalysis models.Overall, this paper provides a comprehensive exploration of deep learning-based steganalysis, highlighting the advantages of CNNs in feature learning for detecting hidden information in images. The proposed approach contributes to the advancement of steganalysis techniques and offers valuable insights for future research in this field.

The proposed method in paper [11] utilizes a deep learning architecture called ResDet, which is based on residual networks. The ResDet model is trained on a large dataset of cover and stego images, with a focus on adaptive JPEG steganography. The authors describe the network architecture in detail, highlighting its ability to capture subtle changes introduced by the steganographic embedding process. To evaluate the effectiveness of ResDet, extensive experiments are conducted on various benchmark datasets that include different steganographic algorithms. The authors compare ResDet with other state-of-the-art steganalysis methods and report the results in terms of detection accuracy. They analyze the performance of ResDet under different scenarios, such as varying embedding rates and steganographic algorithms, to assess its robustness and generalizability.The experimental results demonstrate the superiority of ResDet in detecting adaptive JPEG steganography. The model achieves high detection accuracy and outperforms other existing methods in terms of both false positive rates and false negative rates. The authors attribute this success to the ability of ResDet to effectively capture the subtle modifications introduced by adaptive steganographic techniques.The proposed ResDet model achieves high detection accuracy and outperforms other state-of-the-art methods

[12]Steganography involves cloaking relevant information within other data in order to facilitate safe data exchange. Traditional steganography techniques alter least significant bits (LSB) to conceal data, but it is still difficult to increase their accuracy. The authors stress how crucial it is to take into account the order in which video frames are displayed because this has a big impact on how accurately steganography techniques are used on videos.In order to enhance the hiding process, the authors of this study introduce the usage of a 3D CNN for video steganography, which integrates spatial and temporal information. This strategy is thought to be the pioneering one of its sort. Implementing the suggested strategy on the UCF101 dataset yields results that significantly outperform LSB-based approaches. The authors average a 22.75 bits per second.The use of 3D CNNs in video steganography offers a number of benefits. The suggested method collects more detailed details by combining spatial and temporal data, leading to an increase in concealing capacity and visual quality. The authors emphasise how deep learning methods could improve the accuracy and efficiency of steganography systems.The outcomes of the experiments show how successful the suggested strategy is. The successes point to the promise of 3D CNNs in applications for video steganography. In comparison to conventional LSB-based strategies, the results indicate that the suggested method can offer a more reliable and secure way of concealing information within films.Additionally, the studies' usage of the UCF101 dataset demonstrates how the suggested technique can be used in real-world situations. Although the results are encouraging, the authors are aware that there are still problems. To address potential weaknesses and attacks on the suggested strategy, more research is required. Furthermore, ensuring invisibility to human senses is a critical aspect that needs to be carefully taken into account. Future research can concentrate on strengthening the 3D CNN-based steganography system's resistance to different steganalysis methods.The study report concludes with a novel method for video steganography that makes use of 3D CNNs. When compared to conventional LSB-based algorithms, the integration of spatial and temporal data with deep learning techniques offers significant gains in concealing capacity and video quality. The results of the experiments show that the suggested strategy has the potential to offer a more reliable and efficient way of obfuscating information in videos.

A unique deep learning method is put forth in the paper [13] for identifying steganography in digital photos. The suggested solution makes use of a Siamese convolutional neural network (CNN) architecture, which receives input from two images and generates a similarity score that assesses how similar the images are. A pair of images, one of which is a cover image and the other of which is either a stego image or a cover image, is used to train the network. To teach the network how to distinguish between the two categories of photos is the aim.Three datasets, BOSSBase, BOWS2, and HUGO, which are often used in steganalysis research, are utilised to assess the suggested technique. On all three datasets, the results demonstrate that the suggested method performs better than state-of-the-art techniques. The authors also undertake in-depth tests to examine the effects of several elements on the effectiveness of the suggested technique, including the size of the training dataset, the embedding rate, and the image format.The suggested Siamese CNN method shows promise for identifying steganography in digital photos and can be viewed as an important improvement over the steganalysis techniques already in use.

1.3 Problem Statement

Hiding secret image in another image using convolutional neural networks .

1.4 Applications

Some of the applications of steganography are listed below:

- Secure Communication
- Digital watermarking
- Data hiding
- Covert Intelligence and Espionage
- Digital Forensics
- Linguistic Steganography

- Secure Communication: Steganography offers a way to communicate securely via insecure networks, such as the internet. Sensitive information can be transferred without drawing the attention of listeners or possible attackers by enclosing messages within innocent carriers, such as photos, audio files, or even plain text documents. The concealed message may only be extracted and decoded by authorised receivers who have the necessary decryption equipment or information, protecting the communication's secrecy.

- Digital watermarking: Digital watermarking is the act of inserting undetectable information into digital media files to identify the owner or copyright holder of the content. Steganography is a key component of this process. Unique identities or ownership information can be covertly added to photos, movies, or audio files by using steganographic techniques. The media file's embedded watermark is unaffected by duplication, modification, or distribution, allowing the original author to assert ownership or monitor

unauthorised usage of their intellectual property.

- Data hiding: Steganography may be used to conceal critical or significant information in seemingly harmless files like spreadsheets, text documents, or any other type of digital format. This method is especially helpful in circumstances when using encryption alone can cause unwanted attention or suspicion. The data is made less obvious and less likely to be targeted by unauthorised people by being hidden inside an apparently innocent file. The only people who can retrieve the disguised information are those who are knowledgeable about the steganographic technique utilised and have access to the required instruments.

- Covert Intelligence and Espionage: Steganography has been used for covert communication and information exchange by intelligence organisations and spies for a long time. Agents can communicate safely while avoiding suspicion by hiding hidden communications within seemingly benign files or even physical things. This makes it possible for agents to covertly share private information, strategies, or directives, maintaining the confidentiality of espionage operations.

- Digital Forensics: Steganography has a place in this field since it may be used to both hide evidence and find information that has been concealed. Steganography is a technique that criminals use to conceal unlawful items, including illegal photographs or documents, within seemingly innocent files. Steganalysis techniques are used by forensic professionals to find and recover concealed data, assisting in investigations and offering proof in court cases.

- Linguistic Steganography: Steganography can be used in spoken or written language in addition to digital media. Linguistic steganography is the practise of concealing information in plain text using a variety of methods, such as minute adjustments to phrasing, punctuation, or even voice. Steganography of this kind might be applied to clandestine communication or as a fun exercise for linguists.

1.5 Objectives and Scope of the project

1.5.1 Objectives

Our objective is to :-

- To hide image using CNN so that it is not detectable by unauthorized person.
- To protect the confidentiality of the hidden image by making it difficult for unauthorized individuals to access or extract the information.
- To extract hidden image from a given cover image using neural networks.
- To Create and put into use a steganography system based on CNN.
- To analyse the CNN-based steganography system's performance in terms of image quality, hiding capacity, and resistance to steganalysis attacks.

1.5.2 Scope of the project

- Steganography based on CNN that involves embedding secret data into visuals.
- Assuring the stego image's imperceptibility and visual quality.
- Juggling the capacity and resilience of hidden information.
- Using CNNs to analyse and extract features.
- Adding encryption methods to the mix to improve security.

- Applications include data protection and secure communication.

- Steganalysis is included in the scope to find buried data.

The next chapter will discuss the Requirement analysis of the steganography process.

Chapter 2

REQUIREMENT ANALYSIS

The goal of requirement analysis, a crucial stage in the software development process, is to pinpoint the wants and needs of various stakeholders for a given system. To determine the functional and non-functional needs of the proposed system in the instance of steganography employing CNN, requirement analysis is required. Non-functional requirements pertain to the system's quality and performance, whereas functional requirements outline the features and functionalities of the system.

Convolutional neural networks (CNN) will be used in this project to create a steganography system that can conceal sensitive information in digital photos while preserving the calibre and integrity of the cover image.The project's scope includes defining the system's requirements, utilising CNN to design and develop the steganography system, and assessing the system's performance using a variety of metrics. The system may need to be able to embed and extract data from several image file types, have the ability to conceal a huge quantity of data, and be impervious to steganalysis techniques.

The visual appeal of the cover image, the system's ability to precisely embed and extract data, and the system's resistance to various steganalysis techniques may all be performance measures. The requirement analysis process is essential to ensuring that the created system satisfies stakeholders' needs and expectations and provides the expected results.

2.1 Functional Requirements

- The system should be able to hide data within the images using the CNN algorithm

- The system should be able to extract the hidden data from the images using the same CNN algorithm.

- The system should be compatible with different image formats and sizes to allow users to hide data in any image.

- The system should be designed to ensure the security of the hidden data and to prevent unauthorized access to it.

- The model must be robust to attacks. The model should be able to withstand attacks that attempt to detect or remove the embedded data.

2.2 Non Functional Requirements

- The model should be able to generalize well across different types of cover images, such as natural scenes, textures, or specific domains. It should perform consistently and effectively on various image formats, resolutions, and color spaces.

- Given the secret image and cover image, the container image should be generated within 5 seconds.

- It shall be resistant to distortion caused by image compression, noise addition, cropping, and other common image manipulations, ensuring the hidden image remains intact and retrievable.

2.3 Hardware Requirements

- Processor: For steganography applications, a quick processor is necessary because data concealing and extraction operations can be computationally demanding.

- Graphics card: For steganography applications, especially those involving image processing or machine learning methods, a specialised graphics card can be helpful.

- Memory: For steganography projects to handle substantial volumes of data and images, enough RAM is required.

2.4 Software Requirements

- Programming language:- that support deep neural network language support front he system
- TensorFlow deep learning framework
- Image Processing Libraries: Python Imaging Library) for image handling and manipulation.
- Anaconda, Colab
- Data and image Datasets
- GPU support system

The next chapter will discuss the System Design of the work.

Chapter 3

SYSTEM DESIGN

A number of processes are involved in the system design for steganography utilising CNN, including input data pre-processing, data embedding, model training, and concealed data extraction and detection. The system is first fed with input data, often in the form of an image or media file. Pre-processing methods are then used to clean up and get the data ready for embedding.

Using CNN, which has been trained to determine the best place to embed the data while maintaining imperceptibility, the data is then added to the cover image. After the embedding is finished, the model is trained to find and extract the cover image's hidden data.The system's general design in figure 3.1 attempts to make sure that the hidden data is invisible to human senses and that the system is strong enough to fend off any prospective model-attacks.

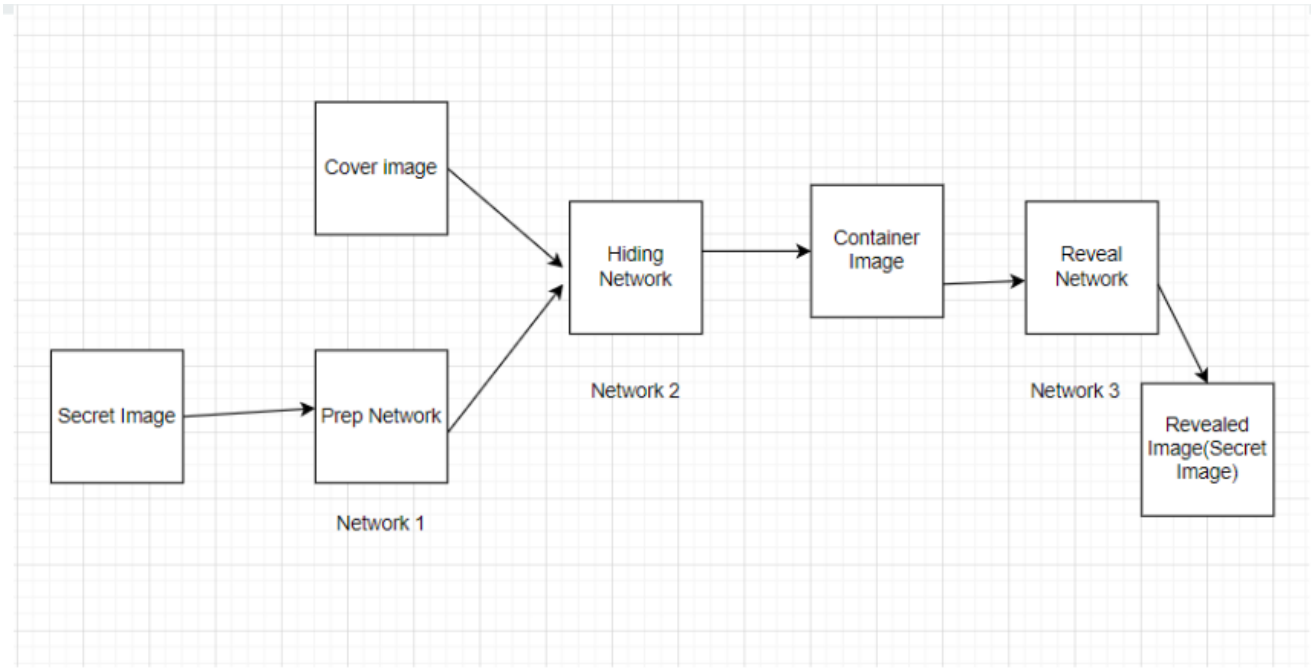

Figure 3.1: System Design of Steganography

3.1 Architecture Design

The architecture of the system will consist of three components trained as a single network; however, it is easiest to describe them individually.

- The cover image and secret message for the hidden network are created by the prep network. The cover image and secret message are inputs, and the programme outputs a preprocessed version of the cover image. The secret message and the preprocessed image are then sent across the concealed network.

- The second/main network, the Hiding Network, takes as input the output of the preparation-network and the cover image, and creates the Container image.The secret message is concealed in the pre-processed cover image using the hidden network. It generates a stego image from the inputs of the secret message and the preprocessed cover image. The stego image should look identical to the cover image and also have the secret message hidden within. The hidden network's architecture is created to strike a balance between reducing the distortion of the cover image and increasing the hidden message's capacity.

- Finally, the right-most network, the Reveal Network, is used by the receiver of the image.The concealed secret message in the stego image must be extracted via the reveal network. It produces the concealed secret message as an output and accepts the stego image as input. Recovering the secret message from the stego image accurately while reducing false positives is the goal of the reveal network's architecture.

  A loss function that penalises cover picture distortions, reduces hidden message error, and ensures that the stego image cannot be separated from the cover image is used to jointly train all three networks. Using backpropagation and stochastic gradient descent, the networks' weights are optimised during the training phase. The networks can be used to encode and decode hidden messages in cover images once they have been trained.

3.2 Data Design (Ex. Database tables or data structures used)

Depending on the particular study endeavour, a different dataset may be used for steganography utilising CNN. The dataset typically contains a sizable number of cover images, which are the images used to conceal the data. These cover images may come from various media files, including photos, digital artwork, or other sorts of media.

The file also contains a collection of hidden information that is utilised to create cover images. Text, photos, and audio files can all be used to represent this hidden information. The dataset is used to train the CNN model to find the best approach to integrate the secret data while maintaining the visual integrity of the cover image.

Data Design for Steganography using CNN involves the representation of data that is used as input to the model for encoding and decoding the secret message. The data consists of two types of images - cover images and secret images. Cover images, which are usually chosen to be similar to the secret image, are the images that will be used as a carrier for the secret message. Secret images are the images that contain the secret message that needs to be hidden in the cover image. The data design involves the following:

- Cover images: The cover images should be of high quality and should have a resolution of at least 256 x 256 pixels. They should be selected in such a way that they are similar to the secret image to avoid detection.

- Secret images: The secret images can be of any size and can be in any format like JPEG, PNG, etc. They should be of high quality to ensure that the message is hidden correctly in the cover image.

- Data preprocessing: Before feeding the data to the CNN model, it needs to be preprocessed. This involves converting the images to arrays of pixel values and normalizing them to values between 0 and 1. The images also need to be resized to a fixed size for consistency

- Data augmentation: Data augmentation techniques like random cropping, flipping, and rotating can be used to generate more data for training the model and to make it more

robust.

- Data splitting: The data needs to be split into training, validation, and testing sets. The training set is used to train the model, the validation set is used to tune the hyperparameters and prevent overfitting, and the testing set is used to evaluate the performance of the model.

Secret images: The secret images can be of any size and can be in any format like JPEG, PNG, etc. They should be of high quality to ensure that the message is hidden correctly in the cover image.

.

3.3 User Interface Design

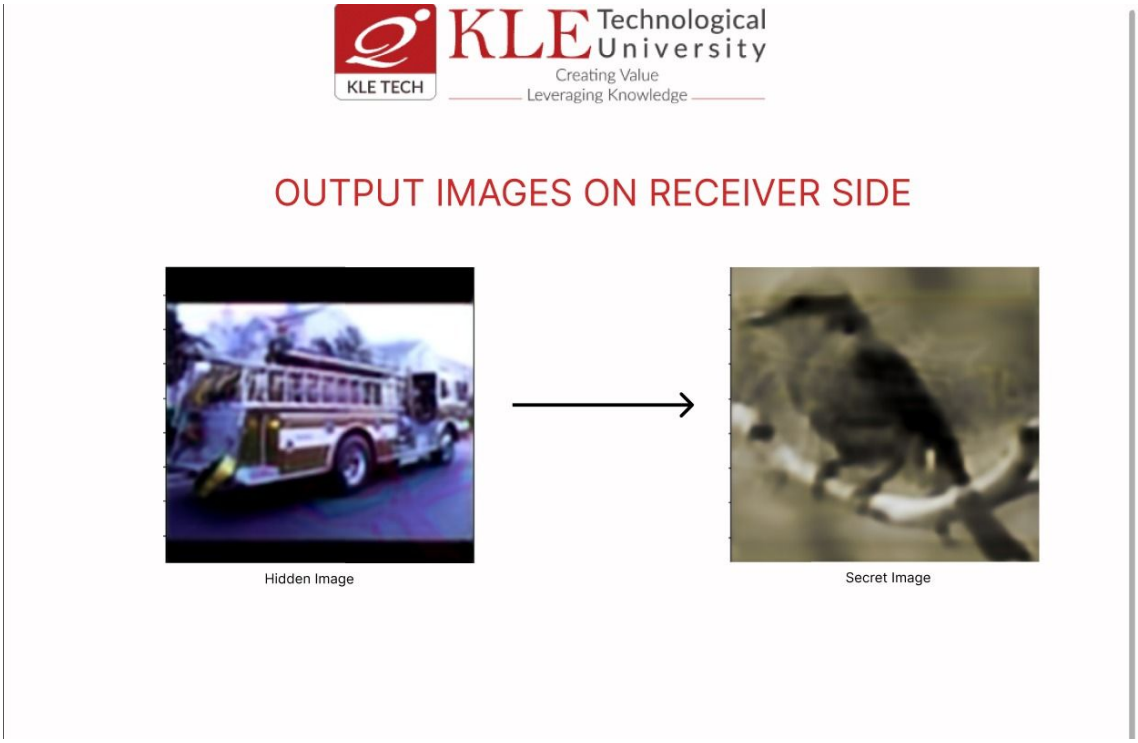

Figure 3.2: System Design of Steganography

The user interface as shown in figure 3.2 of the steganography using CNN project is designed to be user-friendly and easy to navigate. It consists of a simple GUI that allows users to perform various operations related to steganography. The main window of the interface is divided into two main sections: the input section and the output section.

In the input section, users can select the cover image and the secret message that they want to hide. They can either choose to upload the files from their computer or capture the image using the built-in camera of the device. Once the files are selected, the user can click on the "Hide" button to start the process of embedding the secret message into the cover image using the CNN model.

In the output section, the user can view the result of the steganography process. They can view the modified cover image with the hidden message, save the image to their computer, or share it on social media platforms. Additionally, the user can choose to reveal the hidden message from the cover image by selecting the "Reveal" option. This will extract the hidden

message from the cover image using the CNN model and display it on the screen.

The user interface also includes various settings and options that users can customize according to their preferences. For example, users can adjust the compression level of the cover image, select the type of steganography algorithm to use, and specify the encryption method for the hidden message.

Overall, the user interface of the steganography using CNN project provides an intuitive and efficient way for users to perform steganography operations with the help of the powerful CNN model.

The next chapter will discuss the Implementation of the work.

Chapter 4

IMPLEMENTATION

This chapter gives a brief description about implementation details of the system by describing each component with its code skeleton in terms of algorithm.Three phases make up the CNN steganography implementation: prep network, concealed network, and reveal network.The cover image and secret message for the hidden network are created by the prep network. The cover image and secret message are inputs, and the programme outputs a preprocessed version of the cover image. The secret message and the preprocessed image are then sent across the concealed network.The secret message is concealed in the pre-processed cover image using the hidden network. It generates a stego image from the inputs of the secret message and the preprocessed cover image.

The stego image should look identical to the cover image and also have the secret message hidden within. The hidden network's architecture is created to strike a balance between minimising cover image distortion and maximising the capacity of hidden network.The concealed secret message in the stego image must be extracted via the reveal network. It produces the concealed secret message as an output and accepts the stego image as input.

Recovering the secret message from the stego image accurately while reducing false positives is the goal of the reveal network's architecture.A loss function that penalises cover picture distortions, reduces hidden message error, and ensures that the stego image cannot be separated from the cover image is used to jointly train all three networks. Using backpropagation and stochastic gradient descent, the networks' weights are optimised during the training phase. The networks can be used to encode and decode hidden messages in cover images once they have been trained.

4.1 Pre-processing

Various pre-processing methods have been used:

- Normalization: The pictures' pixel values are normalised via the normalize batch function. It divides by the standard deviation values [0.229, 0.224, 0.225] after subtracting the mean values [0.485, 0.456, 0.406] from each pixel. The input data are standardised during this normalisation stage, which also enhances the model's training-phase convergence.

- Denormalization: Normalization's inverse process, denormalization, is carried out using the denormalize batch function. The mean values [0.485, 0.456, 0.406] are added after each pixel value is multiplied by the standard deviation values [0.229, 0.224, 0.225]. The normalised picture data is returned to its original scale for visualisation or additional processing during this denormalization stage.

- Image Loading and Resizing: The "get img batch" method uses the Image.open function to import a random selection of cover and secret pictures from the given files list. If the photos are not already in RGB format, it changes them to that format. The photos are then enlarged to the required size (for example, (224, 224)) while keeping the aspect ratio using ImageOps.fit.

- Scaling and Image Type Conversion: The photos are scaled before being transformed to numpy arrays using np.array. The photos' pixel values are then scaled by multiplying them by 255. The [0, 1] range for pixel values is ensured by this step.

4.2 Prep Network

The figure 4.1 shows the implementation of Prep Network.

- The preparation network takes a tensor containing the secret message as input and returns a tensor that is concatenated with the cover image to form the stego image.
- The preparation network consists of three convolutional branches of sizes 3x3, 4x4, and 5x5.
- Each branch consists of five convolutional layers, each with 50 filters and a ReLU activation function.
- The output from all the three branches are concatenated to form a single tensor.
- Finally, the output of the three convolutional layers is concatenated to form the final tensor that is returned by the function.
- In summary the preparatory network applies multiple convolutional layers with different kernel sizes to the input tensor secret tensor.

```python
import tensorflow as tf

def get_prep_network_op(secret_tensor):
    with tf.variable_scope('prep_net',reuse=tf.AUTO_REUSE):
        c = lambda x, k, n: tf.layers.conv2d(inputs=x, filters=50, kernel_size=k, padding='same', name=n, activation=tf.nn.relu)
        conv_3x3 = c(secret_tensor, 6, "3x3_conv_branch/1")
        for i in range(2, 6):
            conv_3x3 = c(conv_3x3, 6, f"3x3_conv_branch/{i}")
        conv_4x4 = c(secret_tensor, 4, "4x4_conv_branch/1")
        for i in range(2, 6):
            conv_4x4 = c(conv_4x4, 4, f"4x4_conv_branch/{i}")
        conv_5x5 = c(secret_tensor, 5, "5x5_conv_branch/1")
        for i in range(2, 6):
            conv_5x5 = c(conv_5x5, 5, f"5x5_conv_branch/{i}")
        concat_1 = tf.concat([conv_3x3],axis=3,name='concat_1')
        final_5x5 = c(concat_1, 5, "final_5x5")
        final_4x4 = c(concat_1, 4, "final_4x4")
        final_3x3 = c(concat_1, 6, "final_3x3")
        concat_final = tf.concat([final_3x3],axis=3,name='concat_final')
        return concat_final
```

Figure 4.1: Prep Network Algorithm

4.3 The Hiding Network

The figure 4.2 shows the implementation of hiding network

- The function takes in two arguments: cover image and output of prep network
- The hiding layer combines the secret image and cover image by concatenating them along the channel axis (axis 3) and passing the concatenated tensor through a series of convolutional layers
- The purpose of this function is to create a neural network that can be used to hide an image(secret image) within another image (cover image).
- The network architecture consists of three branches, each consisting of five convolutional layers with varying kernel sizes (3x3, 4x4, and 5*5).
- Each branch consists of five convolutional layers, each with 50 filters and a ReLU activation function.
- The Hiding Network applies additional convolutional layers (final 5x5, final 4x4, and final 3x3) to the concatenated tensor obtained from the previous step. These convolutional layers have different kernel sizes (5x5, 4x4, and 3x3) and are followed by ReLU activations.
- Finally, the output of the three convolutional layers is concatenated to form the final tensor that is returned by the function.
- The final output of the network is an image with the same dimensions as the cover image. This output is the hidden image created by the network.

```
def get_hiding_network_op(cover_tensor, prep_output):
    with tf.variable_scope('hide_net'):
        concat_input = tf.concat([cover_tensor, prep_output], axis=3, name='images_features_concat')

        for kernel_size in [6, 5, 4]:
            with tf.variable_scope(f"{kernel_size}x{kernel_size}_conv_branch"):
                conv = concat_input
                for i in range(5):
                    conv = tf.layers.conv2d(inputs=conv, filters=50, kernel_size=kernel_size, padding='same',
                                            name=str(i+1), activation=tf.nn.relu)

        concat_1 = tf.concat([conv], axis=3, name='concat_1')

        conv_5x5 = tf.layers.conv2d(inputs=concat_1, filters=50, kernel_size=5, padding='same',
                                    name="final_5x5", activation=tf.nn.relu)
        conv_4x4 = tf.layers.conv2d(inputs=concat_1, filters=50, kernel_size=4, padding='same',
                                    name="final_4x4", activation=tf.nn.relu)
        conv_3x3 = tf.layers.conv2d(inputs=concat_1, filters=50, kernel_size=6, padding='same',
                                    name="final_3x3", activation=tf.nn.relu)

        concat_final = tf.concat([conv_3x3], axis=3, name='concat_final')
        output = tf.layers.conv2d(inputs=concat_final, filters=3, kernel_size=1, padding='same', name='output')

        return output
```

Figure 4.2: Hiding Network Algorithm

4.4 Reveal Network

The figure 4.3 shows the implementation of Reveal Network

- The function takes in two arguments: container image having both secret image and cover image.]

- The purpose of this network is to identify the presence of hidden information in the container file by detecting any patterns or irregularities that suggest the presence of hidden data.

- By training this network on a dataset of container files with and without hidden data, it can learn to recognize the characteristics of hidden data and reveal them when present.

- The output of this function is a tensor representing the revealed information.

```
def get_reveal_network_op(container_tensor):
    with tf.variable_scope('reveal_net'):
        c = tf.layers.conv2d
        conv_3x3 = c(container_tensor, 50, 6, activation=tf.nn.relu, name="3x3_conv_branch/1", padding='same')
        conv_3x3 = c(conv_3x3, 50, 6, activation=tf.nn.relu, name="3x3_conv_branch/2", padding='same')
        conv_3x3 = c(conv_3x3, 50, 6, activation=tf.nn.relu, name="3x3_conv_branch/3", padding='same')
        conv_3x3 = c(conv_3x3, 50, 6, activation=tf.nn.relu, name="3x3_conv_branch/4", padding='same')
        conv_3x3 = c(conv_3x3, 50, 6, activation=tf.nn.relu, name="3x3_conv_branch/5", padding='same')

        conv_4x4 = c(container_tensor, 50, 4, activation=tf.nn.relu, name="4x4_conv_branch/1", padding='same')
        conv_4x4 = c(conv_4x4, 50, 4, activation=tf.nn.relu, name="4x4_conv_branch/2", padding='same')
        conv_4x4 = c(conv_4x4, 50, 4, activation=tf.nn.relu, name="4x4_conv_branch/3", padding='same')
        conv_4x4 = c(conv_4x4, 50, 4, activation=tf.nn.relu, name="4x4_conv_branch/4", padding='same')
        conv_4x4 = c(conv_4x4, 50, 4, activation=tf.nn.relu, name="4x4_conv_branch/5", padding='same')

        conv_5x5 = c(container_tensor, 50, 5, activation=tf.nn.relu, name="5x5_conv_branch/1", padding='same')
        conv_5x5 = c(conv_5x5, 50, 5, activation=tf.nn.relu, name="5x5_conv_branch/2", padding='same')
        conv_5x5 = c(conv_5x5, 50, 5, activation=tf.nn.relu, name="5x5_conv_branch/3", padding='same')
        conv_5x5 = c(conv_5x5, 50, 5, activation=tf.nn.relu, name="5x5_conv_branch/4", padding='same')
        conv_5x5 = c(conv_5x5, 50, 5, activation=tf.nn.relu, name="5x5_conv_branch/5", padding='same')

        concat_1 = tf.concat([conv_3x3],axis=3,name='concat_1')

        conv_5x5 = c(concat_1, 50, 5, activation=tf.nn.relu, name="final_5x5", padding='same')
        conv_4x4 = c(concat_1, 50, 4, activation=tf.nn.relu, name="final_4x4", padding='same')
        conv_3x3 = c(concat_1, 50, 6, activation=tf.nn.relu, name="final_3x3", padding='same')

        concat_final = tf.concat([conv_3x3],
```

Figure 4.3: Reveal Network Algorithm

The next chapter will discuss the Results of the work.

Chapter 5

RESULTS AND DISCUSSIONS

This chapter discusses on the results of the implemented steganography model.

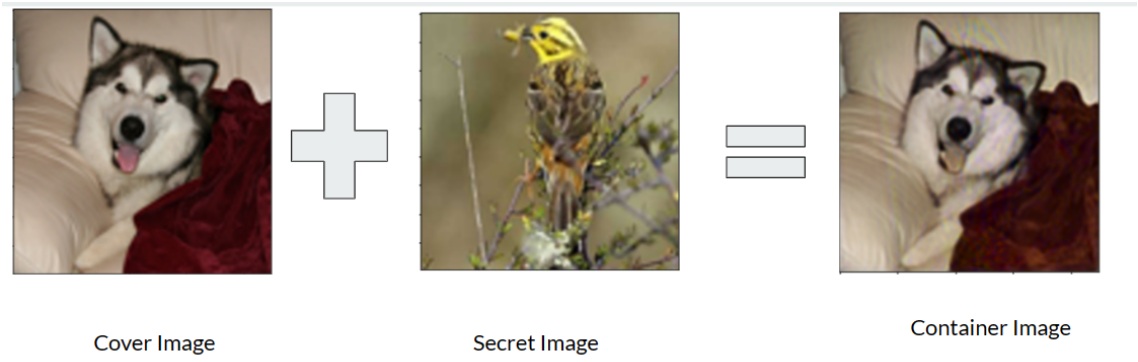

Figure 5.1: Prep Network and Hidden Network

- A secret image is hidden inside a cover image to avoid detection by unauthorized users.

- The secret image is transformed by a prep network to encode useful features for succinct encoding.

- The cover image and transformed secret image are then used as input for a hide network, which creates a container image that conceals the secret image as show in figure 5.1

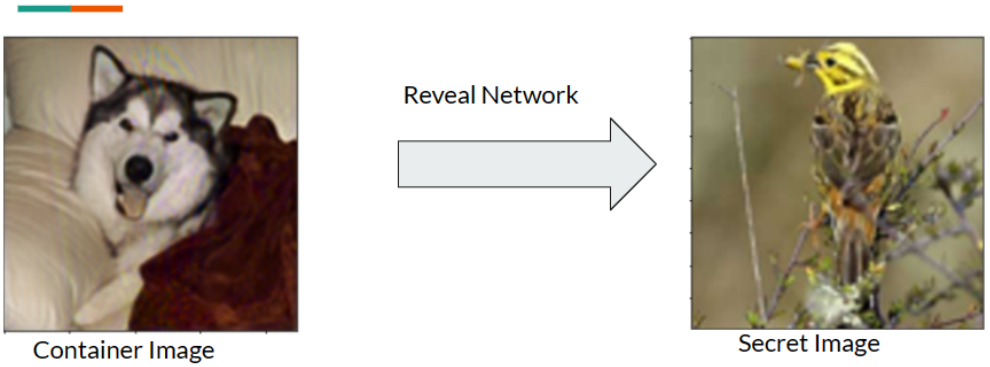

Figure 5.2: Reveal Network

- The Container image is used as input to Reveal Network.
- The Secret Image is extracted from the Container Image as shown in figure 5.2

The next chapter will discuss the conclusions drawn from the implemented model in steganography.

Chapter 6

CONCLUSION AND FUTURE SCOPE

In conclusion, the technique of image steganography using CNN has shown promise in enhancing data security. The architecture employed, consisting of the prep-network, hiding network, and reveal network, has provided a framework for efficient data hiding. While the field is still evolving, recent research has yielded encouraging results, establishing CNN-based steganography as a reliable and effective method for data concealment. However, challenges remain in addressing potential attacks on the CNN model and ensuring imperceptibility to human senses. Further research is needed to enhance reliability and security by incorporating encryption and watermarking methods. The future of steganography lies in the development of more robust and undetectable techniques, capable of withstanding diverse attacks and analysis methods, ultimately enhancing the stealthiness and resilience of hidden information.

REFERENCES

[1] Vijay Kumar, Saloni Laddha, Nitin Dogra Aniket, et al. Steganography techniques using convolutional neural networks. *J. Homepage*, 7:66–73, 2020.

[2] Tao Zhang, Hao Zhang, Ran Wang, and Yunda Wu. A new jpeg image steganalysis technique combining rich model features and convolutional neural networks. *Mathematical Biosciences and Engineering*, 16(5):4069–4081, 2019.

[3] Chi-Kwong Chan and Lee-Ming Cheng. Hiding data in images by simple lsb substitution. *Pattern recognition*, 37(3):469–474, 2004.

[4] Weixuan Tang, Bin Li, Mauro Barni, Jin Li, and Jiwu Huang. An automatic cost learning framework for image steganography using deep reinforcement learning. *IEEE Transactions on Information Forensics and Security*, 16:952–967, 2020.

[5] Chuan Qin, Weiming Zhang, Hang Zhou, Jiayang Liu, Yuan He, and Nenghai Yu. Robustness enhancement against adversarial steganography via steganalyzer outputs. *Journal of Information Security and Applications*, 68:103252, 2022.

[6] Vijay Kumar, Pankaj Rao, and Ashish Choudhary. Image steganography analysis based on deep learning. *Review of Computer Engineering Studies*, 7(1):1–5, 2020.

[7] Ying Zou, Ge Zhang, and Leian Liu. Research on image steganography analysis based on deep learning. *Journal of Visual Communication and Image Representation*, 60:266–275, 2019.

[8] Weixuan Tang, Bin Li, Shunquan Tan, Mauro Barni, and Jiwu Huang. Cnn-based adversarial embedding for image steganography. *IEEE Transactions on Information Forensics and Security*, 14(8):2074–2087, 2019.

[9] Guoliang Xie, Jinchang Ren, Stephen Marshall, Huimin Zhao, and Rui Li. A novel gradient-guided post-processing method for adaptive image steganography. *Signal Processing*, 203:108813, 2023.

[10] Ru Zhang, Hao Dong, Zhen Yang, Wenbo Ying, and Jianyi Liu. A cnn based visual audio steganography model. In *Artificial Intelligence and Security: 8th International Conference, ICAIS 2022, Qinghai, China, July 15–20, 2022, Proceedings, Part I*, pages 431–442. Springer, 2022.

[11] Nagham Hamid, Balasem Salem Sumait, Bilal Ibrahim Bakri, and Osamah Al-Qershi. Enhancing visual quality of spatial image steganography using squeezenet deep learning network. *Multimedia Tools and Applications*, 80(28-29):36093–36109, 2021.

[12] Li Li, Weiming Zhang, Chuan Qin, Kejiang Chen, Wenbo Zhou, and Nenghai Yu. Adversarial batch image steganography against cnn-based pooled steganalysis. *Signal Processing*, 181:107920, 2021.

[13] Xiaosa Huang, Shilin Wang, Tanfeng Sun, Gongshen Liu, and Xiang Lin. Steganalysis of adaptive jpeg steganography based on resdet. In *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 549–553. IEEE, 2018.

[14] Wen-Hsiang Tsai Da-Chun Wu. A steganographic method for images by pixel-value differencing. *Pattern Recognition Letters*, 24:1613–1626, 2003.

[15] Lingamallu Naga Srinivasu and Vijayaraghavan Veeramani. Cnn based "text in image" steganography using slice encryption algorithm and lwt. *Optik*, 265:169398, 2022.

[16] Jindou Liu, Zhaohong Li, Xinghao Jiang, and Zhenzhen Zhang. A high-performance cnn-applied hevc steganography based on diamond-coded pu partition modes. *IEEE Transactions on Multimedia*, 24:2084–2097, 2021.

[17] Chuan Qin, Weiming Zhang, Xiaoyi Dong, Hongyue Zha, and Nenghai Yu. Adversarial steganography based on sparse cover enhancement. *Journal of Visual Communication and Image Representation*, 80:103325, 2021.

[18] Chao Yuan, Hongxia Wang, Peisong He, Jie Luo, and Bin Li. Gan-based image steganography for enhancing security via adversarial attack and pixel-wise deep fusion. *Multimedia Tools and Applications*, 81(5):6681–6701, 2022.

[19] Aayush Mishra, Suraj Kumar, Aditya Nigam, and Saiful Islam. Vstegnet: Video steganography network using spatio-temporal features and micro-bottleneck. In *BMVC*, volume 274, 2019.

[20] Varsha Himthani, Vijaypal Singh Dhaka, Manjit Kaur, Geeta Rani, Meet Oza, and Heung-No Lee. Comparative performance assessment of deep learning based image steganography techniques. *Scientific Reports*, 12(1):16895, 2022.

[21] Jianhua Yang, Kai Liu, Xiangui Kang, Edward K Wong, and Yun-Qing Shi. Spatial image steganography based on generative adversarial network. *arXiv preprint arXiv:1804.07939*, 2018.

[22] Zhili Xiang, Jun Sang, Qian Zhang, Bin Cai, Xiaofeng Xia, and Weiqun Wu. A new convolutional neural network-based steganalysis method for content-adaptive image steganography in the spatial domain. *IEEE Access*, 8:47013–47020, 2020.

[23] Ruohan Meng, Zhili Zhou, Qi Cui, Xingming Sun, and Chengsheng Yuan. A novel steganography scheme combining coverless information hiding and steganography. *Journal of Information Hiding and Privacy Protection*, 1(1):43, 2019.

[24] Ashish Choudhary Vijay Kumar, Pankaj Rao. Image steganography analysis based on deep learning. *Review of Computer Engineering studies*, 7(1):43, 2020.