QATAR UNIVERSITY

COLLEGE OF ENGINEERING

IMAGE STEGANOGRAPHY USING DEEP LEARNING METHODS TO DETECT COVERT COMMUNICATION IN UNTRUSTED CHANNELS

BY

NANDHINI SUBRAMANIAN

A Thesis Submitted to

the Faculty of the College of Engineering

in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Computing

January 2021

COMMITTEE PAGE

The members of the Committee approve the Thesis of
Nandhini Subramanian defended on 24/11/2020.

_____
Dr.Somaya Al-Maadeed
Thesis/Dissertation Supervisor

_____
Dr. Camel Tanougast
Committee Member

_____
Dr. Elias Edward Yaacoub
Committee Member

_____
Dr. Nizar Barah Zorba
Committee Member

Approved:

_____
Khalid Kamal Naji, Dean, College of Engineering

ABSTRACT

SUBRAMANIAN, NANDHINI., Masters: January : [2021:], Masters of Science in Computing

Title: Image Steganography Using Deep Learning Methods to Detect Covert Communication in Untrusted Channels

Supervisor of Thesis: Somaya, Al-Maadeed.

Media has become a primary medium of communication with the help of the constantly evolving technology. Social media like Facebook, YouTube, Twitter, WhatsApp, and other sites have become a platform for exchanging audio, video, and text messages. This has left them vulnerable to attacks which makes it essential to protect the confidential messages sent over the media channel. Image steganography is the procedure used for camouflaging a secret image in a cover image. In contrast, the method of detecting and extracting the secret information from the stego image is called steganalysis. Steganography can be used positively to secure the data transmission process. On the other hand, it can be used adversely by hackers, criminals, and covert operators for the secret exchange of messages.

This work aims to develop a steganography model to embed the secret image and steganalysis tool to extract the embedded secret image. In recent times, deep learning methods have gained popularity and are widely used in the field of steganography. In this work, a unique auto encoder-decoder with a deep convolutional neural network is proposed. Training and testing are done on a subset of the COCO, CelebA, and ImageNet dataset. To evaluate the proposed method, Peak Signal-to-Noise

Ratio (PSNR) and Mean Squared Error (MSE) metrics are used. The proposed method has proved to achieve higher invisibility, security, and robustness.

DEDICATION

To my kids and my parents.

To my husband, my pillar of support. Thanks for holding me when I fell and helping me rise above and beyond.

ACKNOWLEDGMENTS

My sincere thanks to my husband, kids, and parents for understanding and supporting me throughout this journey. My biggest thanks to my supervisor, Dr. Somaya Al-Maadeed. Without her, this thesis would not have been possible. Thanks for believing in me and letting me grow. A special mention to Dr.Omar Elharrouss for his constant guidance. Last but not least, thanks to all my friends for putting up with me during all the ups and downs.

This work was made possible by NPRP11S-0113-180276 from the Qatar National Research Fund (a member of the Qatar Foundation). The findings achieved herein are solely the responsibility of the author.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Technology has shaped our lives and its application in our day-to-day life is undeniable. With the constant scientific contribution from the research community, technology is evolving regularly. Technological development has become a measure for a nation's success and other aspects of social development. Specifically, the media has undergone a complete makeover from printed papers to video hosting websites. People have started using media as the primary source of communication. Social media has become a platform to share photos, videos, audio among people. It has helped to break the physical distance and made the exchange of digital media possible. Not only social communication but media platforms are also used for marketing by companies, reviewing on customer service, influencers for creating content on social awareness, publishing news digitally by the newspapers, and so on. Apart from social media, email communication has become the go-to method for communicating officially and unofficially. Numerous video hosting websites like YouTube, TikTok, and others are easily available for the general population. Technology is enabling us and making our lives significant, however on the other side of the coin, privacy and security are compromised.

Digital media are transferred through untrusted communication channels and hence it can be easily tampered with; or it can be used deliberately by terrorists, hackers, and other people with bad intent, to communicate the secret meeting locations. Inversely, it is essential for government officials, police, and security officers to communicate confidential information, company secrets in an effective way that cannot be intercepted and tampered with. Information security methods are necessary to transfer the data securely and also for interception and decoding any illicit secret communications.

Information hiding is a topic which deals with hiding important confidential information from attackers and third parties. Cryptography, watermarking, and steganography are the prevailing information hiding techniques [1], [2]. Cryptography deals with converting plaintext into a ciphertext which is not readable. Cryptography consists of two algorithms – encryption and decryption. At the sending end, the encryption algorithm is used to produce the ciphertext from the plain text. At the receiving end, the decryption algorithm is deployed to decrypt the plain text from the ciphertext [3].

The process of embedding confidential information in any digital media is called watermarking. The embedding can be a symbol, logo which can be used to identify the ownership, preserve the copyrights [4], [5]. The process of hiding any confidential and secret information inside digital media in plain sight is steganography. The hidden message is not visible to the human eye which makes it immune to attacks [6]. Figure 1 explains the different information hiding techniques. Image steganography which is highlighted in red in figure 1 is the technique that will be used throughout this work.

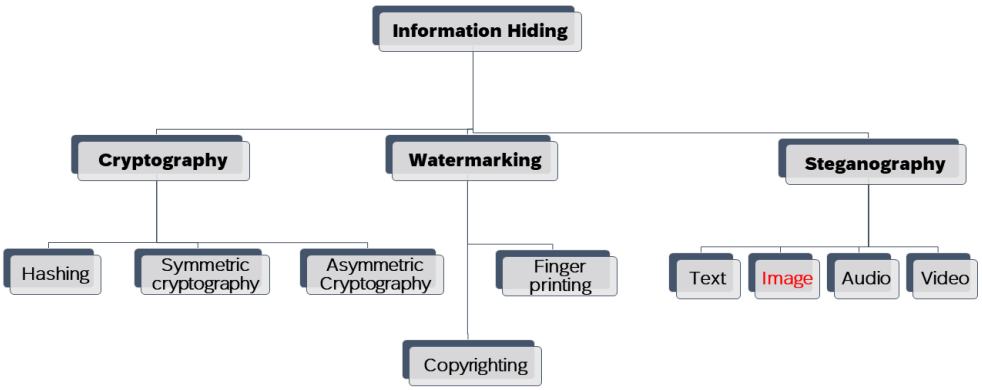

Figure 1. Different types of the information hiding techniques.

1.1 Image Steganography

The direct translation of the Greek word Steganography is “cover” for stego and “writing” for graphy [7]. Steganography is used to protect confidential information by hiding it inside digital media. The existence of the hidden message is known exclusively to the sending end and the receiving end. The digital media used as the cover can be text, image, audio, video, or digital files. Image steganography is the sub-branch of steganography in which image is used as the cover media. In other words, the image is used as the cover under which secret information can be text or another image is hidden. The process of detecting the existence and uncovering the secret information from the generated steganography image is image steganalysis. Usually, in covert communication, the steganography algorithm is placed at the sending end and the steganalysis algorithm is placed at the receiving end [8].

1.2 Steganography Vs Cryptography

Though the purpose of steganography and cryptography may look similar, there is a subtle difference between them [9]. Steganography and cryptography are used in communicating secret information between two parties; however, the hidden message is not visible to human eyes in steganography. This is not the case in cryptography, though the hidden message is not visible, the encrypted ciphertext shows the presence of a secret message. In steganography, the structure of the original message is preserved while in cryptography, the whole structure of the original message is altered. Cryptography can be applied only on text, but steganography can be used on text, audio, images, and video. Cryptography is commonly used, unlike steganography. Table 1 compares and contrasts the key difference between steganography and cryptography.

Table 1. Difference Between Cryptography and Steganography.

Criteria	Cryptography	Steganography
Definition	Hiding secret information in undecipherable and non-readable form	Hiding secret information that is imperceptible to human sight
Translation	Secret writing	Cover writing
No. of inputs	One	Minimum of two
Visibility	Though not breakable, visible to human eyes	Not visible to human eyes
Counterpart	Cryptanalysis	Steganalysis
Original data	Structure is altered	Not altered
Secret information	Mostly Text	Text, audio, video, files, and images
Key	Key is essential	Optional to increase the security
Popularity	Very familiar	Emerging

1.3 Motivation

Initially, the correspondence happened by tattooing in a clean shaved head of slaves to communicate the secret information [10]. With the advent of technology, digital media especially images became common. To hide secret messages inside the images, a popular technique called the Least Significant Bit (LSB) substitution was used. In LSB methods, the least significant bits in the input image which are prone to noises are identified and modified to hide the binary form of the secret text. The main

concern here is the hiding capacity, security, and robustness of the method. For example, the hiding capacity in LSB methods is between 0.1 and 0.4 bits per pixel (bpp) which is meager.

In recent times, deep learning methods are extensively used for many applications like computer vision and other image processing applications. Though image steganography is in a bidding stage of research, it has benefited from deep learning methods. General Adversarial Networks (GAN) which are known for its performance in image reconstruction methods are used in image steganography. Apart from GAN, encoder-decoder based convolutional neural networks (CNN) are also used. The main motivation of this thesis is to apply deep learning methods, mainly the autoencoder-decoder network to perform end-to-end image steganography and steganalysis. The hiding capacity of the proposed method is increased to 1 bpp since the cover and secret images used are 3 channels (RGB).

1.4 Research Questions

The answers to the below mentioned research questions will be focused primarily,

- Can a deep convolutional auto encoder-decoder network be used for image steganography and steganalysis?
- Is it possible to design and develop a lightweight model architecture without any complicated layers to perform the image steganography and steganalysis?
- Does the proposed method without any complicated architecture provide better results than other related researches?

1.5 Research Aim and Objective

End-to-end image steganography and steganalysis using a simple, lightweight autoencoder network is the main aim of this research.

This study has the below mentioned objectives,

- To design and implement a simple, lightweight deep convolutional autoencoder network which can take two image inputs to produce two image outputs.
- Define a new loss function to deal with two image inputs and two image outputs.
- Train and test the proposed model across three different datasets.
- Evaluate the proposed method. Since the evaluation is based on image similarity, MSE and PSNR are used.
- Compare the results from the proposed method against other methods.

Summary

In this chapter, the evolution of the image steganography methods over the decades, the motivation of the research to use deep learning methods for image steganography is elaborated. The research questions, aim, and objectives of the thesis are also outlined in this chapter.

CHAPTER 2: BACKGROUND

In this chapter, necessary background details required to fully understand the concepts of the proposed method are presented. This chapter includes the working principle of the convolutional neural network with different hyperparameters that the networks use. The auto encoder-decoder network is the building block of the proposed method and is used heavily in the upcoming sections. The applications of the auto encoder-decoder network in different fields are also described to understand the power of the AE network.

2.1 Convolutional Neural Networks

CNN is a branch of deep learning method specifically artificial neural networks which is inspired by the living creatures' natural visual perception mechanism [5]. CNNs are nothing but stacked multi-layered neural networks. There are three major categories of layers: convolutional, pooling, and the dense and fully connected layers. The first layer is an input layer where the width, height, and depth of the input image is used. Right after the input layer, convolutional layers are defined with the number of filters, filter window size, stride, padding, and activation as the parameters. Convolutional layers are used to extract some meaningful feature maps for the input location by taking the weighted sum. The feature map is then passed through an activation function and bias is added to form the output. Usually, a ReLU activation is used which is calculated using $x = \max(0, y)$.

The size of the output from the convolutional layers is reduced using the pooling layer. As the model increases with increasing filters in the convolution layer, the output dimensionality also increases exponentially which makes it hard for the computers to handle. Pooling layers are added to reduce the dimensions to make it easy for computation and sometimes to suppress the noises. The pooling layer can be max

pooling, average pooling, global average pooling, or spatial pooling. The most commonly used pooling layer is a max-pooling layer. A single-array feature vector flattened from the output is then fed to a fully connected layer. Finally, a classification layer is defined with some activation functions like sigmoid, softmax, or tanh. The number of classes is specified in this layer and aggregates the features extracted into class scores.

Batch Normalization layers are applied after the input layer or after the activation layers to standardize the learning process and reduce the training time. Another important parameter is the loss function which summarizes the error in the predictions made in the training and validation sets during training. The loss is fed to the CNN model after each epoch to enhance the learning process.

2.2 Auto encoder-decoder

Auto encoder-decoder networks are a variant of the convolutional neural networks used for unsupervised learning with symmetric structure. From the training examples, autoencoders are used to construct an output image similar to the given input image. Encoder networks are mainly used in image compression and reconstruction tasks. Autoencoder networks are similar to principal components analysis (PCA) in dealing with compressing the input. However, nonlinearity is introduced in autoencoder using the activation functions in each layer, unlike PCA which represents linear transformations of the input. Suppose I represents the input image, the main aim of the autoencoder is to produce an output I_o such that $I_o = I$. Autoencoder networks have a symmetrical structure with the number of layers in the encoder (including the middle layer) is equal to the number of layers in the decoder.

There are three main parts in any autoencoder network, namely, encoder, latent space, and the decoder.

The encoder is the contracting part with a decreasing number of filters in an autoencoder network. The encoder part of the network is responsible for downsampling the input image by extracting features. Encoder aims at converting the input image into a feature vector representation. The structure of the encoder is similar to a convolutional neural network.

Latent Space is also known as the bottleneck layer is sandwiched between the layers of an encoder and the layers of the decoder. The latent space representation is the compressed version of the input image and is the outcome of the encoder. Latent space is given to the decoder to reconstruct an output image which is close to the input image from the compressed features.

The decoder is the expanding part with increasing filters in the autoencoder. The decoder takes the compressed feature vectors produced by the encoder and upsamples it to reconstruct the output image. Figure 2 shows the basic architecture of an autoencoder network.

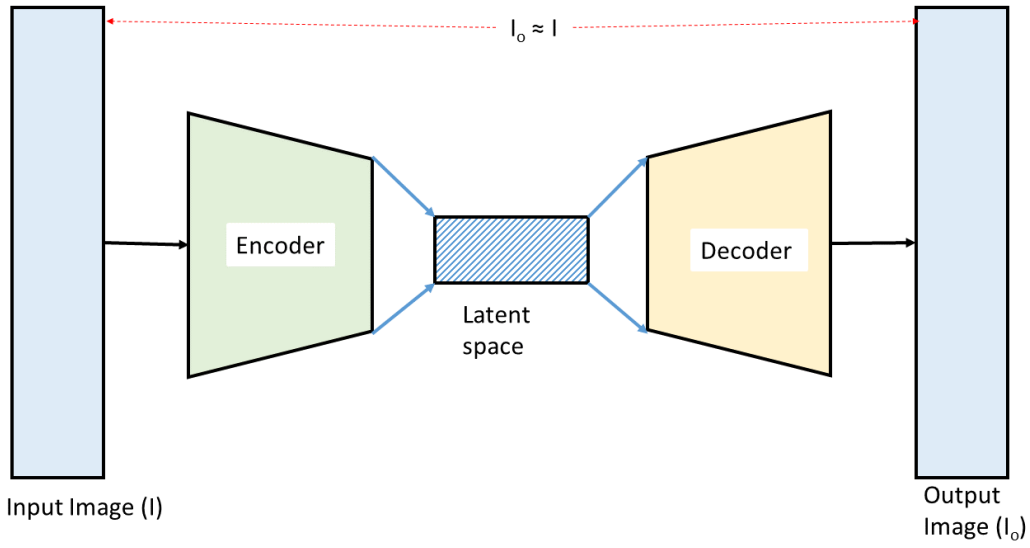

Figure 2. Basic architecture of autoencoder-decoder network.

In any given image, not all the pixel values or features are useful. In other words, each image will have numerous useless, and redundant features that will not be helpful during training. Rather these features will increase the computational time, storage, and memory. To avoid such cases, the autoencoder is used to represent the image with compressed values that have only useful features. The reconstructed image from the autoencoders can be used as input to other models.

Some of the interesting points to note in autoencoder are, the autoencoder networks are data-specific, which means that the trained model can perform well only on inputs that are similar to the training sample. The autoencoder networks are prone to heavy distortions and loss, that is, the reconstructed output will suffer from compression loss. Though unsupervised learning, the learning process happens automatically without any requirement for help from humans or labeled data. The model can learn to produce outputs from the given input images without any labeling and hence is helpful to perform well in specific tasks rather than generally.

2.3 Different Variations of Auto encoder-decoder

There are different variants of the autoencoder networks available. Some of them are described with explanations on their working below. The different variations of the autoencoder network are given in figure 3.

1. Undercomplete Autoencoder (UAE) networks are made up of fewer nodes in the hidden layer when compared to the input layer. The main purpose is to capture only the important features from the input by penalizing the network for the errors in the reconstruction of the input. UAE does not require any regularization but is prone to overfitting.
2. Sparse Autoencoder (SAE) networks are made up of more nodes in the hidden layer when compared to the first layer (input), unlike UAE. Sparsity is

introduced in the hidden layer to prevent the decoder from copying the input. The sparsity penalty is applied to the network along with the reconstruction error.

3. In Contractive Autoencoder networks**,** the Frobenius norm of the Jacobian matrix which is the summation of the square of all the elements in the hidden layer is calculated. A comparison is made between the input and the calculated value. Based on the calculated value, the loss term is penalized to learn robust features from the input that can help the decoder to minimize the error during reconstruction.
4. Convolutional filters are used in the Convolutional Autoencoder networks in encoder as well as a decoder. It is the most commonly used form of autoencoder in recent times.
5. Denoising autoencoder networks**:** A random noise is added to the input before passing it to the network in denoising autoencoder networks. Adding noise is another way of helping the network to learn better feature representation and avoids copying input in the output.
6. The input image's latent feature representation is used in the Variation Autoencoder networks (VAE). Probabilistic distribution of the input is given by the encoder and the decoder constructs the output as close to the input from taking samples from the probability distribution given by the encoder.
7. Deep Autoencoder networks (DAE) consists of the encoder and the decoder made up of a deep belief network. The encoder and the decoder have the same number of layers which can vary from 4 to 8. A convolutional layer is used at the end of the decoder to output the image. One such deep autoencoder model is proposed in this thesis for the application of image steganography.

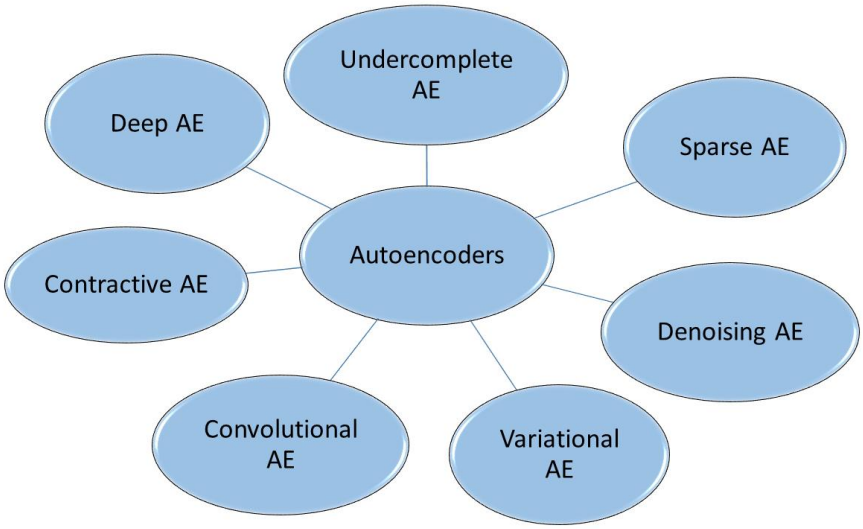

Figure 3. Different variations of autoencoder networks.

2.4 Applications of autoencoders

Some of the applications of the auto encoder-decoder network in different fields and the most popular encoder network used are given in this section. Figure 4 shows the different applications of AE in a diagrammatic view.

- Image classification: For transforming and compressing the training samples into another form before using them for classification is done by autoencoder. One such example is the classification of medical images, say, using nuclei to detect breast cancer as the authors did in [11] using stacked DAEs.
- Dimensionality Reduction: With the introduction of mobile edge computing, dimensionality reduction can be helpful in the transfer of data between the end devices and the cloud servers [12]. This will help in reducing the computational time and reduce the storage overhead at the cloud servers.
- Colorization of Images: To color images that are black/white to respective colors based on the available information from the input images appropriately.

U-Net is a famous encoder-decoder based neural network which is used in image colorization [13].

- Image denoising: Autoencoder networks are used to remove noises and blurs from RGB images. Denoising autoencoder networks are best suited to perform image denoising. A combination of deep and convolutional autoencoder is used for image denoising in [14].
- Digital Watermarking: watermarks which are copyrights are embedding in digital images and video by software to claim ownership. Autoencoder-decoder networks are heavily used in this field [5].
- Feature Extraction: The latent space representation of the input image obtained from the encoder is rich in useful features. The encoder part of the autoencoder can be used to extract features that can be further used by other models for classification. A variation autoencoder network (VAE) is used in [15] for extracting features from ECG signals which are then used to produce augmented and synthetic ECGs.

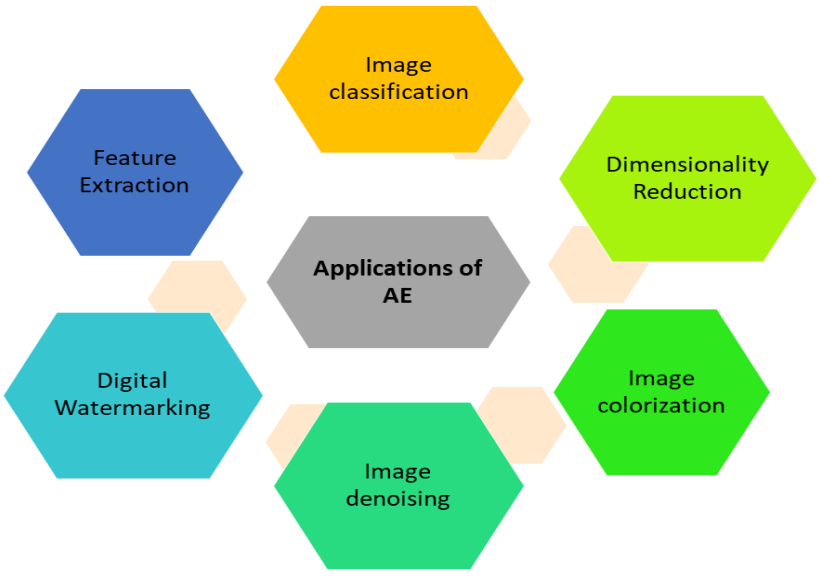

Figure 4. Applications of AE.

Summary

In this chapter, the working principle of the convolutional neural network, different hyperparameters that are used in the networks are briefed. The definition of autoencoder networks, structural elements, properties, different variations, and applications of autoencoder networks are also included.

CHAPTER 3: RELATED WORK

The art of hiding secret information inside a cover image that is not perceptible to human eyes is image steganography. Image steganography and steganalysis are two methods that have gained popularity in recent times. Image steganography methods have evolved historically from traditional methods to using deep learning methods. In this section, related works that are aimed at performing image steganography is described. The methods available are broadly three categories, namely, traditional steganography methods, CNN-based steganography methods, and GAN-based steganography methods. A brief review of all the existing methods under each category is studied in this chapter.

3.1 Traditional Steganography Methods

Traditionally, the Least Significant Bits (LSB) substitution method is the most often opted for image steganography. Image steganography requires two images. One image which acts as the carrier of the secret which is the cover image. Secondly, the secret information can be cipher text, plain text, digital file, image, or video which will be hidden in the bitstream of the cover image. Image is a representation of intensity variations in the form of pixels. Images are represented either in 24-bit or 8-bit format. In the LSB method, either text or image is chosen as secret media and the cover is also an image. If the secret media chosen is text information, the secret text is converted into binary form. The cover image which is usually natural scene imagery is combed window by window to find the least significant bits in the noisy area. The least significant bits from the cover image is modified with the binary bits from the secret message in such a way that the hidden secret text is imperceptible. It is assumed that the resolution of the cover image is high and exploiting the three least significant bits

for every byte in the image will not reduce the precision or arouse any suspicion [16] and [17].

Ever since the introduction of LSB methods, different variations of these techniques have been used for image steganography. For example, a Huffman encoding of the secret message is performed and the encoding is ingrained in the cover image using the LSB method [18]. The input images used in [3] is grayscale. The secret image is smaller than the cover image. Similar to LSB, yet another commonly used traditional method is Pixel Value Differencing (PVD). In the smooth region, a lesser number of bits and in the edges higher number of bits of the secret message is hidden in PVD techniques. LSB and PVD deal with images in the spatial domain [19]. A combination of the LSB and PVD method is used in [20]. LSB is applied on 2 least bits and PVD in particular, Quotient Value Differencing is applied to other bits. LSB technique has been used even in quantum images in [20]. Information hiding on quantum images with modification of the direction technique is proposed in [21], [22]. The cover quantum image is grouped into N pixels and every secret bit is included in every pixel group of the cover image [7]. An image steganography method using k-LSB techniques is proposed in [7] and entropy filters are used for steganalysis along with the image enhancement method. Local Binary Patterns are also to perform coverless steganography for selecting the features [23].

The major shortcoming in the traditional image steganography method is the capacity. The hiding capacity in the traditional method is very low. Trying to hide more information by tweaking a greater number of bits in the cover may expose the hidden secret information. The security and robustness of the traditional method are also minimum. Since the hiding happens by statistically exploiting the pixel values, steganalysis can be easily done by reverse engineering. This will affect the security of

the method. The quality of the extracted secret information may also be subsided affecting the robustness. Yet another drawback is that the media of secret communication is mostly text. Even if an image is used, only grayscale images are used. Hiding the pixel values of the three-channel RGB image inside another three-channel secret image can get quite difficult in traditional methods.

3.2 CNN-based Steganography Methods

Deep learning is extensively used in applications like computer vision [24] and other tasks dealing with images like face recognition [25], object detection and tracking [26], and gesture and action recognition [27]. Ever since the bloom of the artificial intelligence field, deep learning is the go-to method because of their enhanced performance. Unlike traditional methods and machine learning methods, deep learning algorithms do not depend on manually picked features. With the right configuration and parameters, deep learning methods can learn the best features on their own. Deep learning methods are widely discussed and implemented in the field of image steganography. Since image steganography is like any other image reconstruction, auto encoder-decoder architecture is generally implemented. U-Net and Xu-net are the most popular networks used. However, in image steganography methods that use the convolutional neural network base, there are two input images - the cover and secret image.

Most of the methods in literature have designed and implemented end-to-end steganography and steganalysis method using the CNN architecture. Some methods have used already existing popular networks or proposed new architecture. In both the approaches, the way the input images are concatenated and given as input vary. Another common thing in most of the methods is that an encoder-decoder architecture is used for steganography with a pre-processing module that can help in merging the two input

images. Another CNN model architecture is used as steganalyzer to separate the ingrained secret image from the stego image. One such example can be seen in [8], where a prep-network prepares the input images before passing them into the hiding network. Finally, a reveal network is used to obtain the secret image from the reconstructed container image. A variation of [8] can be seen in [28]. Instead of a passing RGB cover and secret image. The single-channel (Y channel) secret image is embedded in the cover image's Y channel. The output from the hiding network is concatenated with the Cr and Cb channel from the original input image to form the YCrCb cover image.

StegNet [29] and [30], with an embedding and decoding structure is implemented to embed and extract the secret image. One important thing to note is the usage of the depth-wise separable convolutional layer for concatenating the two input images channel-wise. An encoder decoder network is used for both hiding the secret images and extracting the payload also [31]. A u-net based architecture is used in [32] for image steganography. The concatenated version of input images (cover and secret image) is given to the encoder part of the u-net to create the encoded version of the stego image. The decoder part is used as the extraction network to uncover the secret image. A scheme similar to [8] can be seen in [32]. However, there is a subtle difference in the network and in the way cover and secret images are given as input. A preprocessing network is used in [8] whereas the inputs are concatenated in [32]. A similar hiding network and reveal network-based workflow can be seen in [33].

A different approach to image steganography using the convolutional neural network is implemented in [34] and [35]. A styling network that takes a styling image in addition to the cover image to develop the stego image which is completely different from the original cover image is used in [34]. The stego output image not only contains

the secret image embedded but also has a different style than the cover image. A pixel-wise CNN is used in [35] to obtain the pixel distribution of the cover image. The secret image's pixels are distributed across the cover image's pixels obtained from the pixelCNN by reduced sampling.

Though the hiding capacity which is an issue with most of the traditional method is solved. The capacity is increased at the cost of the storage space, memory, and computation time because of the complexity of the models.

3.3 GAN-based Steganography Methods

General Adversarial Networks (GAN) [36] are a part of the deep learning models majorly used for image construction. GAN consists of two networks that compete against each other to improve the overall efficiency of the model. The generator is used to produce an output image that looks like the input image. Discriminator takes the generated image by the generator and classifies it as either real or fake. The feedback is given back to the generator to improve the output image quality. Game theory is used in the adversarial process of image generation. There are different variations of the GAN available. Some of them are Wasserstein GAN (WGAN) [37], Loss Sensitive GAN (LSGAN) and WGAN-With Penalty are two variants of WGAN, Deep Convolutional GAN (DCGAN), StackedGAN [38], CycleGAN [39], Conditional GAN (CGAN) [40], Auxiliary classifier GAN (AC-GAN) [41], InfoGAN [42]. Not only CNNs but also, Recurrent Neural Networks (RNN) are used as the base architecture in GAN [43].

A few additional components are added to the general GAN network structure to customize it to work for image steganography. one such customization can be observed in Steganography GAN (SGAN) [44]. In addition to the generator and discriminator, a steganalyzer is added in Steganography GAN (SGAN) [44].

- A generator, G, to take the input image and the secret information to produce the stego image.
- A discriminator, D, to classify the generated image as stego or not.
- A steganalyzer, S, to uncover the secret message from the generated stego image.

DCGAN is used as the base architecture in [44] with LeakyReLU activation and batch normalization in steganalyzer, ReLU, and TanH in generator and Softmax activation in discriminator. Similarly, three-part steganography using WGAN is proposed in [45] and [46] and named Secure Steganography GAN (SSGAN). Cycle GAN which is famous for unpaired image-to-image translation is prevailing in the image steganography research field. A slight modification to cycleGAN is done in [47], [48], [49] and [50] to perform image steganography. The input images are given in parallel to the cycleGAN generator to output the stego image. The discriminator is used to classify the generated image as normal or stego. Intriguingly, [47] has used cycle GAN to hide medical information of the patients while passing it through untrusted channels in an IoT based cloud computing system. Steganalyzer is replaced with an embedding simulator for developing the modification map from the probabilistic map produced by the generator in [51] and [52]. ACGAN architecture is used in [43], and [44]. First, word segmentation and image database for each word segmentation are developed. From the secret information given, the image database is looked up-to to choose a cover image. The grayscale secret image is hidden using the GAN based encoding network into one channel of the cover image. A decoding network ids used to uncover the hidden grayscale image [55].

A two-way sender-receiver scheme is proposed in [56], [57] where the sender end has the steganography method and the receiving end has the steganalysis method.

Another approach is the coverless steganography methods like in [58], [59], and [53]. In coverless steganography methods, the cover image is generated from a random seed value or text-based information. The generated image acts as the cover to conceal the secret message and at the receiving end, it is decoded. A cryptography-inspired framework can be seen in [60] and [61], where Alice is the generator and Bob is the discriminator and Eve is the steganalyzer. Figure 5 gives an overall view of the steganography GAN and Alice, Bob, and Eve based steganography framework combined. The generator, discriminator, and steganalyzer use one of the GAN variants as the base architecture.

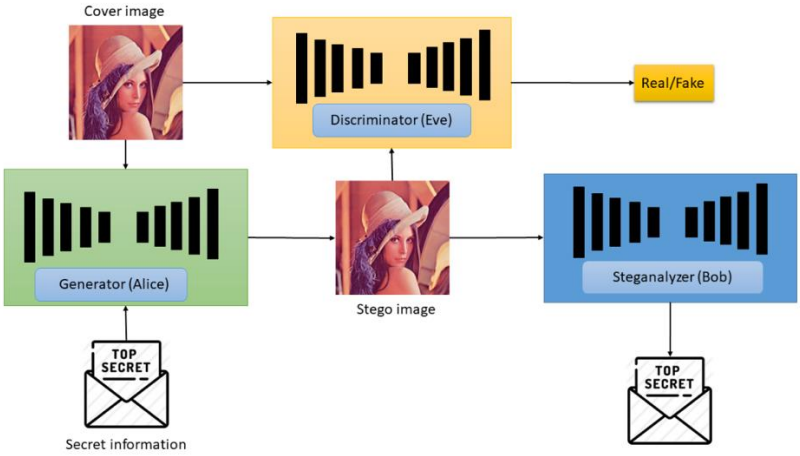

Figure 5. General overview of the GAN-based steganography method.

Summary

In this chapter, all the related works in the image steganography are elaborated. The existing works are grouped into three categories, traditional, CNN-based, and GAN-based image steganography methods. A literature survey covering all three categories are briefed in this chapter.

CHAPTER 4: METHODOLOGY

This chapter will describe in detail the datasets used, the architecture of the proposed model, custom hyperparameters, hardware, and the software specification, train and test dataset split, and the metrics used to evaluate the method.

4.1 Datasets

Data is the crux and core for training any deep learning algorithm. The quantity and the quality of the dataset utilized decide the performance of the model ultimately. It is very essential to choose the dataset used for training the proposed model wisely. There is only one benchmark dataset that can be found for image steganography, BOSSBase [62]. However, the images are grayscale images and hence cannot be used in the proposed system. After careful review from the state-of-the-art papers, three datasets are used. Three different datasets used are – ImageNet [63], COCO [64], and CelebA [65]. A detailed description of the datasets is given below.

4.1.1 ImageNet

ImageNet [63] is a vast dataset with over 15 million images under 80K classes annotated by human labelers. Around 1000 images per class which are nouns from the WordNet dictionary are collected to form the dataset. Not only vast but also, diverse. Original images are not copyrighted, and URLs of the images are given for download. ImageNet dataset is available in four different resolutions, 8x8, 16x16, 32x32, and 64x64 apart from the original resolution of the images. Keras library has tiny ImageNet publicly available and in-built. Tiny ImageNet has images of very low resolution, say, 64 x 64. Images present in the ImageNet datasets are of arbitrary sizes and resolutions. However, based on the input size of the CNN architecture, the images are resized to a fixed size. ImageNet is mainly used for object detection and localization, image

classification. Some examples of the images from the ImageNet dataset can be found in figure 6.

Figure 6. Samples from ImageNet dataset.

4.1.2 COCO

Common Objects in Context (COCO) [64] dataset contains 328K images, 2.5 million images captioned under 91 different categories, captured in various angles, backgrounds, and cameras. COCO is mainly used in object detection, segmentation, and image classification tasks. Images are naturally captured from day-to-day life scenes. Images are mostly objects under the “things” and “stuff” categories. Similar to the ImageNet dataset, images of arbitrary sizes are present in the COCO dataset also. Ground truth files consist of the class labels for object recognition and image classification, segmentation mask for object segmentation, instance spotting, and natural scene segmentation. A few samples from the COCO dataset are given in figure 7.

Figure 7. Examples from COCO dataset.

4.1.3 CelebA

Unlike the other two datasets [63], [64], CelebA [65] dataset consists of celebrity face images captured under different angles, locations, and backgrounds. CelebA dataset contains 200 Million face images and 40 annotated attributes. Images in the CelebA dataset contain different attributes like oval face, long eyebrows, faces with glasses, wearing hats, and different hairstyles. This dataset is used for face recognition, face detection, and attributes localization, face synthesis, and editing tasks. There is no benchmark dataset in image steganography, mostly because it is an unsupervised learning task. Existing datasets use for other topics are modified to fit into the needs of image steganography. Though the above-mentioned datasets are not used for steganography, they are the most commonly used in the field of steganography. Table 2 summarizes the important aspects of the datasets along with the purpose for which the datasets are utilized commonly. Figure 8 gives some samples from CelebA.

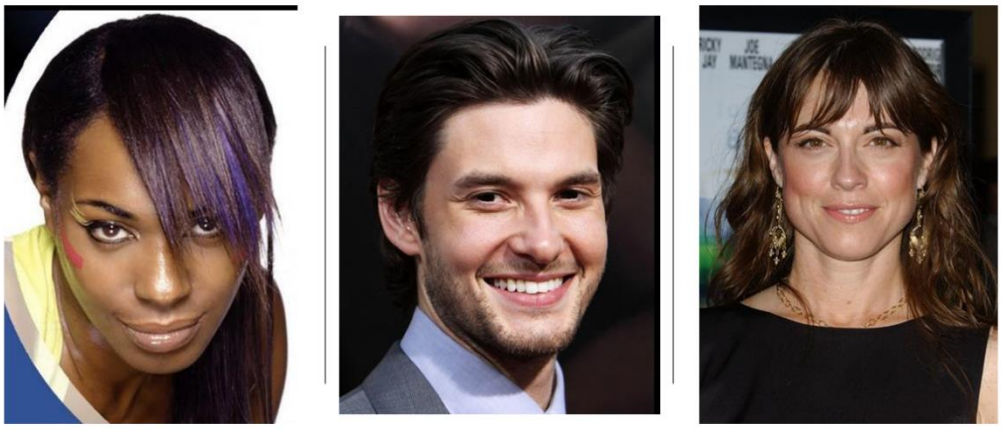

Figure 8. Samples from CelebA dataset.

Table 2. Summary on Details About the Dataset.

Dataset	Total Images	Image Size	Purpose
ImageNet	15 M	Arbitrary	Object detection and localization, image classification
COCO	328 K	Arbitrary	Image classification, Object recognition and segmentation, instance spotting and natural scene segmentation
CelebA	200 M	Arbitrary	Face recognition, face detection and attributes localization, face synthesis and editing tasks

4.2 Proposed Model Architecture

A unique auto encoder-decoder model is proposed to generate a steganography image from the given cover and secret images input. The overall architecture of the proposed model can be divided into three networks, namely, the preprocessing network, the embedding network, and the extraction network. Figure 9 represents the high-level picture of the proposed model. A detailed description of the three networks is given below.

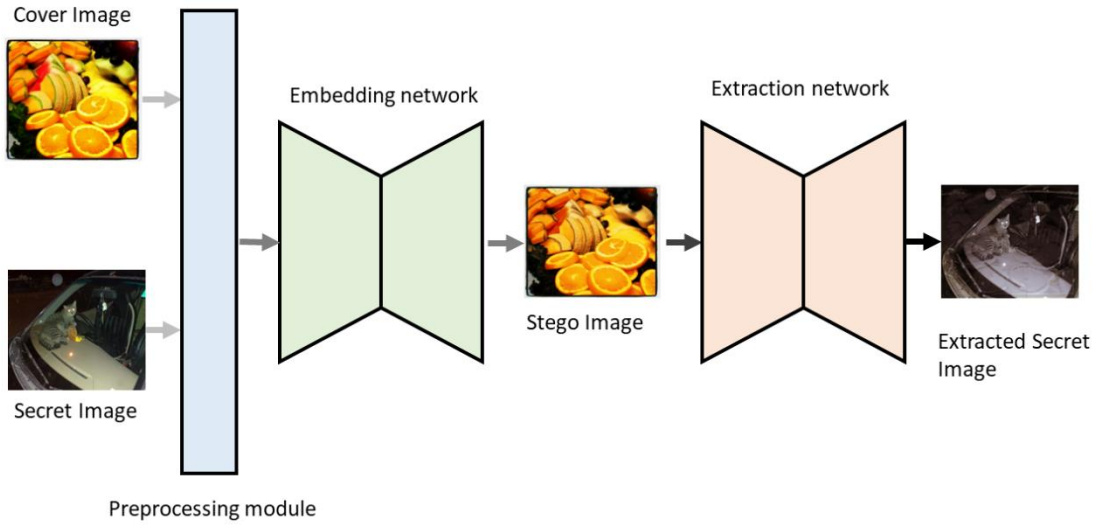

Figure 9. Overall workflow of the proposed method.

4.2.1 Preprocessing Network

The image inputs (cover and secret) are given in parallel and the expected output is also an image (steganography image). There are different ways of concatenating the input images before passing it to the model for producing the stego image. In the proposed model, features are extracted consecutively from cover and secret images. The extracted features are given to the concatenation layer and the output is the

concatenated feature vector of the two images. The concatenated features from the cover and the secret images are then passed to the embedding network to construct the steganography image that looks like the cover image. Irrespective of the original size of the images, the input images are resized to 256 x 256 as the shape of the input layer in the preprocessing network is 256 x 256. The cover image and secret image is processed through three convolutional layers independently. The number of filters used is 8, 16, and 32 with filter size 3 x 3. ReLU activation is used to introduce linearity in each convolution layer. A concatenation layer is introduced to concatenate the feature vectors obtained. The final output from the preprocessing network is the concatenated feature vector of the cover and the secret input images.

4.2.2 Embedding Network

The embedding network is the model solely responsible for converting the concatenated feature vector into a reconstructed image that is proximal to the cover image. As mentioned in chapter 2, an encoder-decoder network contains three components, an encoder, latent space, and a decoder. Part of the encoder network is used for preprocessing and the other part is used to create an image with the secret image embedded. The latent space produced by the encoder is the compressed version of the input image fused with every pixel of the secret image. The decoder section of the embedding network outputs the steganography image from the compressed latent space. The size of the cover image and secret image should be the same. Every pixel from the secret image is distributed across every pixel of the cover image to produce a resultant image that has the secret image hidden but looks like the cover image. Two convolution layers with filter size 3 x 3 and ReLU activation is used in the encoder part of the model. The number of filters in the encoder part is 64 and 128. The decoder has 5 convolution layers with a decreasing number of filters – 128, 64, 32, 16, and 8.

Finally, a convolutional layer with 3 filters is applied to output the steganography image. Figure 10 shows the full architecture of the preprocessing and embedding networks and details on the convolutional layers.

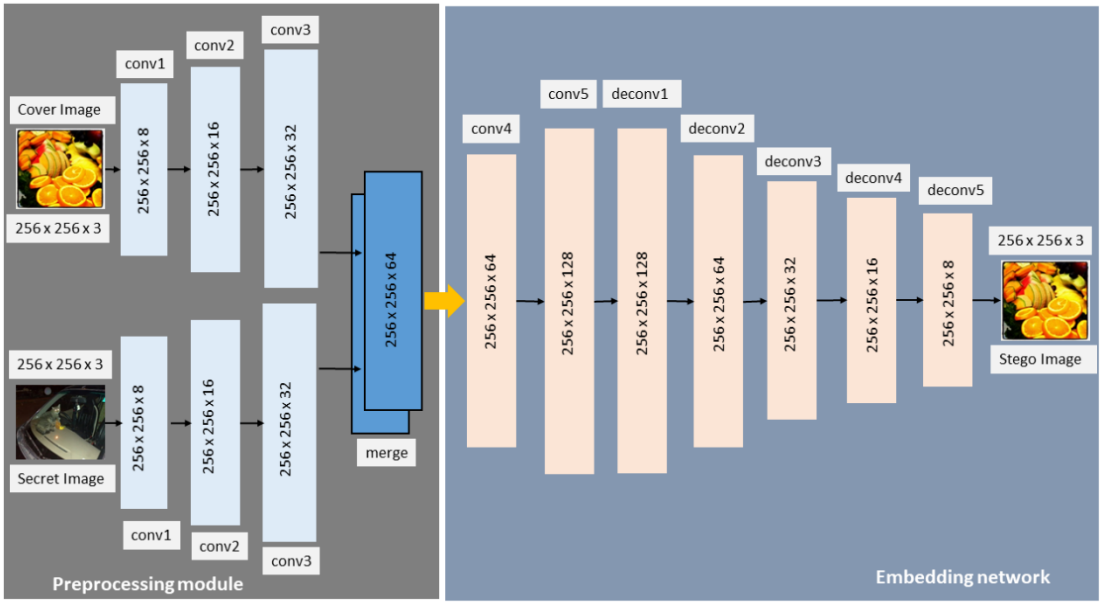

Figure 10. Model architecture of preprocessing and embedding network.

4.2.3 Extraction Network

The extraction network acts as the steganalyzer and decodes the secret image that is camouflaged in the steganography image. The embedding and extraction network have similar architecture. However, the functionalities of the embedding and the extraction network are different. The extraction network takes the steganography image to obtain the embedded secret image. The encoder part of the extraction network consists of 5 convolution layers for downsampling the input image to a latent space vector. The number of filters used in the encoder is 8, 16, 32, 64, and 128 with ReLU activation. The decoder has a decreasing number of filters – 128, 64, 32, 16, and 8 to

upscale the latent space vector to a complete image. A convolutional layer with 3 filters is used to construct and output the secret and cover image concatenated together. The extraction model's architecture is presented in figure 11.

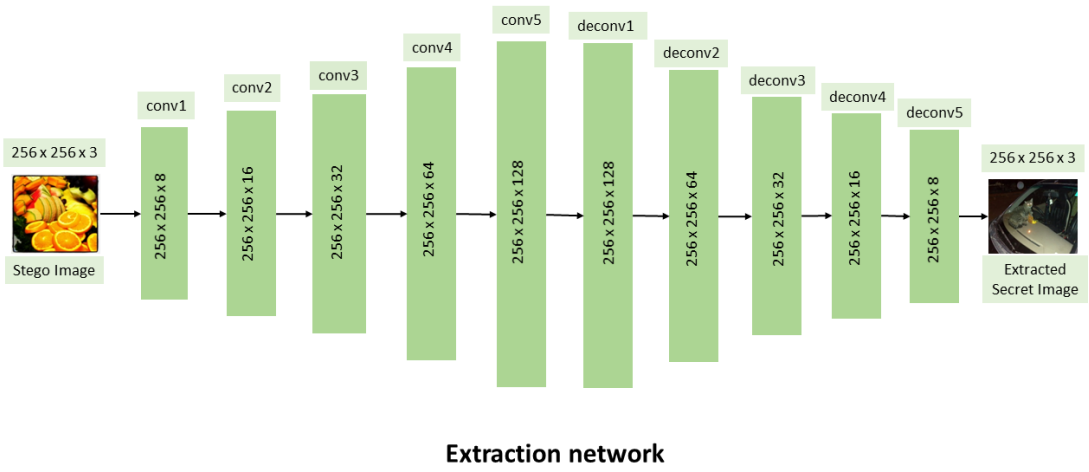

Figure 11. Model architecture of the extraction network.

4.3 Customized Hyper-parameters

Image steganography is an unsupervised learning method similar to the image reconstruction task with a slight difference in terms of the number of inputs and outputs. Unlike image denoising or other image reconstruction tasks, image steganography takes two input images and gives two output images. So, the traditional and ready-available loss functions may not suit the image steganography task. A loss function is an aggregate of the cover distortion loss and the secret distortion loss is customized. The cover distortion loss is the loss between the original cover image and the image output of the embedding network. In the same way, the secret distortion loss is the loss between

the input secret image and the output image of the extraction network. The sum of the cover distortion loss and the secret distortion loss gives the final overall loss. The calculated loss is then given back to the model during training to minimize it for better performance. Different weightage for the cover and secret distortion loss is given before calculating the final loss. For the proposed application, the cover distortion loss is given a weightage of 1, and 0.3 is given for the secret distortion loss, since steganography gets higher importance than steganalysis.

Let's suppose, i is the cover image and i′ is the reconstructed cover image with the secret image generated by the embedding network. h is the secret image and h′ is the extracted secret image by the extraction network. The loss function has to be customized in such a way that it will help the model to optimize the learning function. Loss is a feedback measure given back to the model while training in each epoch as a measure of how well the model is performing through backpropagation.

The loss of the embedding network,L_{emb}, is given by equation 1 and the loss of the extraction network,L_{ext} , is given by equation 2. Finally, the overall loss, L, is calculated using equation 3.

$$L_{emb} = \|i - i'\| \qquad \text{(Equation 1)}$$

$$L_{ext} = \|h - h'\| \qquad \text{(Equation 2)}$$

$$L = L_{emb} + \alpha * L_{ext} = \|i - i'\| + \alpha * \|h - h'\| \text{ (Equation 3),}$$

Where α is the error adjustment and is fixed to 0.3.

Experiments are conducted for values of 0.3, 0.6 and 0.9. Increasing the value of α increased the loss and 0.3 value produced optimal loss value. The value of 0.3 gave the best results. Hence the value is fixed at 0.3. The embedding network's loss function is given back to the embedding network and the overall loss is given to the extraction network to minimize the distortion of the extracted secret image.

4.4 Experimental Setup

In this sub-section, details on the hardware and software specifications used for the experiments are outlined. The training, validation, and test split of the datasets used for training, validating, and testing the proposed method is also described here.

4.4.1 Hardware Specifications

Image steganography takes two input images to produce two output images. 256 x 256 is the image size of both images. The customized datasets used for training, validating, and testing the proposed model contains at least 12000 images in total. Though the proposed system is not complex, the number of parameters the system has to store is high. For example, the total number of trainable parameters in the preprocessing module and embedding model together is approximately 369K. Table 3 gives the layer name and the number of parameters for each layer. Normal CPU cannot handle the huge number of parameters and the data. GPU computing machine with Intel Core i7 and NVIDIA GEFORCE graphic card is used.

4.4.2 Software Specifications

The whole module was developed with Python 3.7 environment along with Keras library. Matplotlib library is used to write and display the images from the datasets. All the datasets used contains images of arbitrary sizes. However, since the input shape of the model accepts only 256 x 256 size, the images were read and resized using the imread and imresize function of the skimage.io library.

4.4.3 Data Split

Datasets chosen for the experiments are popular and huge. The number of images in each of them is in millions. However, for the experiments, this many data were not necessary, and so, 45000 image pairs for cover and secret image input are chosen at random for training. Another 1000 image pairs are chosen and again at

random for testing that does not overlap with the training set. From the 5000 training images, 20% is taken for validation. Introducing a validation set is of paramount importance since it can help the model to validate itself on its performance in each epoch of the training. In this research work, 1000 images out of 45000 training images are used as a validation set.

Table 3. Parameter Details of Preprocessing and Embedding Network.

Network	Layer	Output Shape	# of param
Preprocessing module	Input(cover)	(256, 256, 3)	0
	Input(secret)	(256, 256, 3)	0
	Conv1	(256, 256, 8)	224
	Conv2	(256, 256, 16)	1168
	Conv3	(256, 256, 32)	4640
	Merge	(256, 256, 64)	4640
Embedding network	Conv4	(256, 256, 64)	36928
	Conv5	(256, 256, 128)	73856
	Deconv1	(256, 256, 128)	147584
	Deconv2	(256, 256, 64)	73792
	Deconv3	(256, 256, 32)	18464
	Deconv4	(256, 256, 16)	4624
	Deconv5	(256, 256, 8)	1160
	Output	(256,256,3)	219

4.5 Evaluation Metrics

Evaluation is crucial in understanding the effectiveness and efficiency of the developed method. In the case of image steganography, four main characteristics have to be evaluated, namely - security, robustness, perceptibility, and capacity.

4.5.1 Capacity

Capacity is a metric to measure the amount of secret media that is hidden inside the cover image with minimum distortion. Since the inputs are images of the same size, the capacity is 1. If L is the length of the secret information (text) and C, H and W represent the number of channels, the height, and the width of the cover image. The formula to calculate capacity is given in equation 4. Since the secret media is an image, W, H, and C of the secret image is aggregated to get the L value.

$$Capacity = \frac{L}{W*H*C} \qquad \text{(Equation 4)}$$

4.5.2 Security and Robustness

Security means the ability of the model to provide provable security to the hidden data. In other words, the hidden secret data is not available to attackers. It can be accessed only by the intended authorized users.

Robustness refers to the extent to which the secret media is embedded and retrieved without any loss of information. The secret information should be communicated across the users without any loss.

To measure the security and the robustness, the Peak-to-Noise Ratio (PSNR) measure is calculated. PSNR is a measure to calculate the distortion and similarities between two images and is commonly used in image reconstruction tasks. Firstly, Mean Squared Error (MSE) is measured first and from the calculated MSE, PSNR is computed. Higher values of PSNR indicate that the two images are closer to each other. For the proposed method, two PSNR values are calculated, the embedding PSNR and

the extraction PSNR. The embedding PSNR is calculated between the input image and the output image generated of the embedding network. The extraction PSNR is calculated between the secret image and the output image from the extraction network. The equation for calculating the MSE is given in equation 5 and the equation to compute the PSNR is given in equation 6.

$$MSE = \frac{\sum_{R,C}[I_1(r,c) - I_2(r,c)\,]^2}{R*C} \qquad \text{(Equation 5),}$$

R, C is the total rows and columns in the input images I_1 and I_2. Both the input images in comparison have to be of the same size to calculate MSE.

$$PSNR = 10 * \log_{10} \frac{E^2}{MSE} \qquad \text{(Equation 6),}$$

The fluctuations in the images are given by E which is a fixed value decided based on the input image type. E is 1 for double-precision floating images.

4.5.2 Perceptibility

Perceptibility is the capability of the method to hide secret information that is not visible to the human eyes. Humans should not be able to interpret the secret information hidden. This measure is more related to the visibility of the result images. Result images from the embedding network and the extraction network are added to show that the secret information hidden is not visible. Figure 12, 13, and 14 given in chapter 5 demonstrates the result images along with the original images to show the perceptibility of the proposed model.

Summary

In this chapter, the architecture of the proposed model, the datasets choice, the hyper-parameters, hardware and software specifications, and the evaluation metrics for evaluating the model are described.

CHAPTER 5: RESULTS AND DISCUSSION

The results and the inferences made from the experimental results are given in this chapter. Apart from the detailed reporting of the results, the results obtained are conducted with related works in this field, and the observations made are recorded in this chapter.

5.1 Results and Discussion

MSE, PSNR of the embedding network, and the extraction network are given in table 4 for each dataset. The higher value of PSNR proves the security and the robustness of the proposed model. Image results obtained on COCO, CelebA, and ImageNet are given in figures 12, 13, and 14 respectively. The image results demonstrate the perceptibility of the model.

Table 4. Results of the Proposed Method.

Dataset	Network	MSE	PSNR
COCO	Embedding	44.01	31.96
	Extraction	105.37	27.90
ImageNet	Embedding	51.97	34.55
	Extraction	104.92	27.93
CelebA	Embedding	41.15	32.26
	Extraction	105.10	27.92

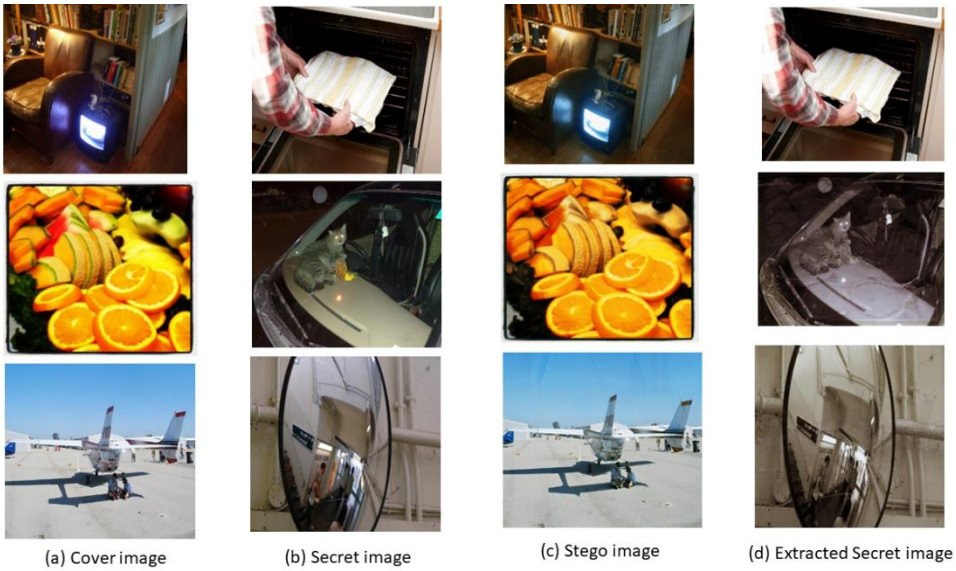

Figure 12. Image results of COCO dataset.

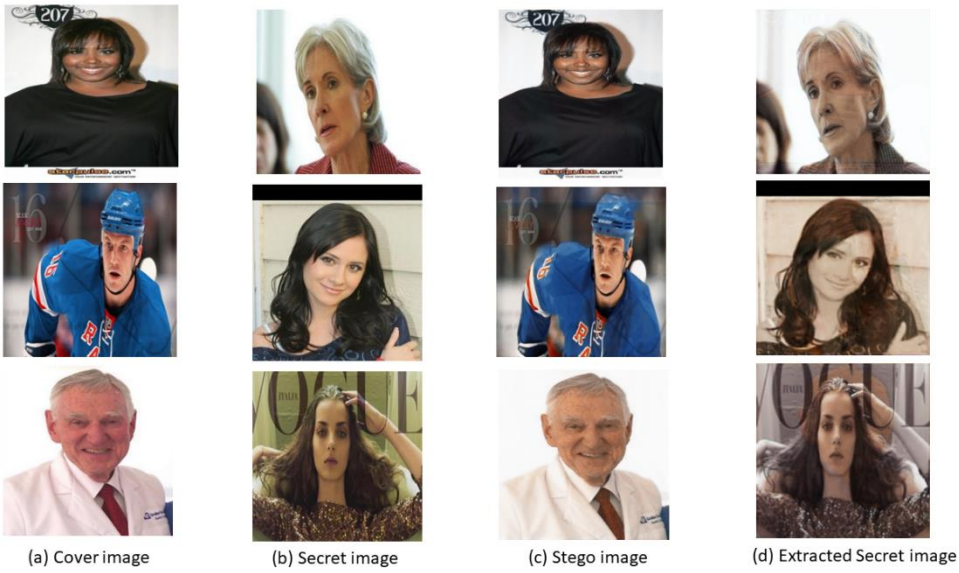

Figure 13. Image results from the CelebA dataset.

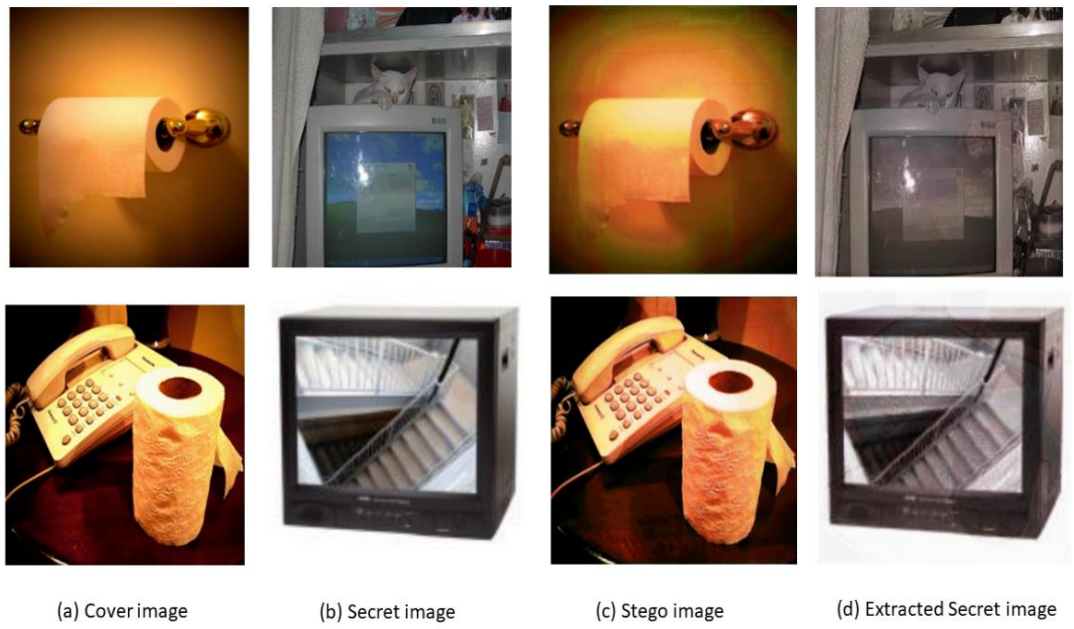

Figure 14. Samples image results of the ImageNet dataset.

Time is another important factor to be evaluated for any algorithm. Sometimes, based on the criticality, a compromise between efficiency and time is made. A critical comparison of the time taken for training the embedding network on all the three datasets along with the computation time taken for the trained model to load and generate a single stego image is made in the bar chart shown in figure 15. A similar comparison for the extraction network is made in figure 16. For the time analysis, the proposed embedding network takes twice the amount of time compared to the extraction network. This is because the embedding network has to process two RGB images, whereas extraction network has to process one image only.

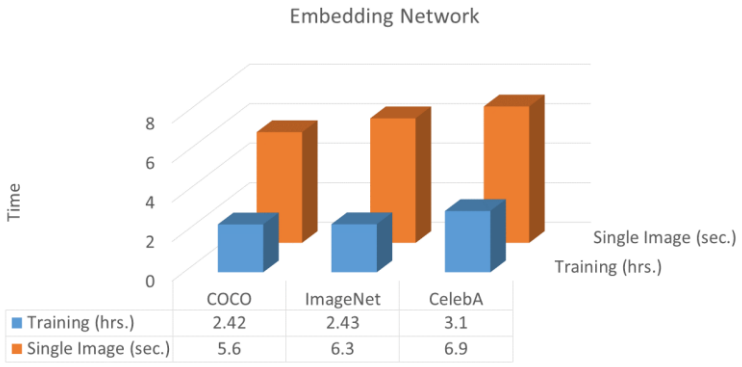

Figure 15. Critical time analysis on embedding network.

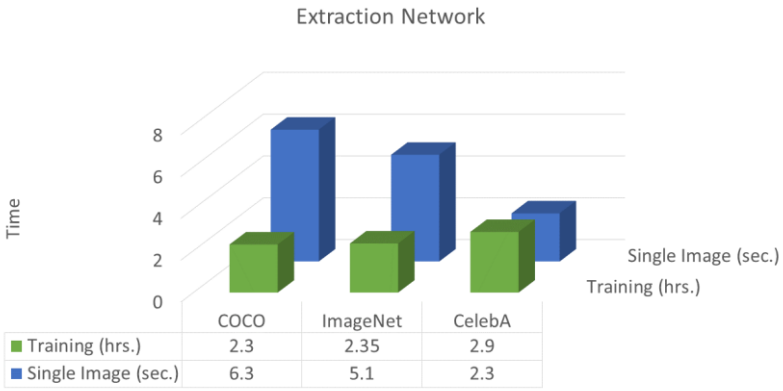

Figure 16. Critical time analysis of the extraction network.

For comparing the ability of the proposed model with [32] and traditional steganography methods, different experiments are conducted. ImageNet dataset has been used in [32]. The authors in [32] have used four image pairs for testing their model. The image results of the four image pairs used in [32] are acquired and used for testing the proposed model is given in figures 17, 18, and 19 along with the histogram comparison. Table 5 represents the comparison of the PSNR values of the proposed

method with other state-of-the-art methods. To compare the proposed method, Rehman's method [31], Zhang's method [28] and Chen's method [55] are used and all the methods in comparison use the ImageNet datasets. From the table 5, the proposed method has the best PSNR value.

Table 5. Comparison of the Proposed Method with Other Methods.

Method	PSNR
Rehman's method [31]	29.6
Zhang's method [28]	33.92
Chen's method [55]	34.07
Proposed Method	**34.55**

It is a common practice in the image steganography research field to test the hypothesis on common and popular images like the Lena, Airplane, Baboon, and Peppers. Similarly, these common images are passed as input to the method, and the PSNR results obtained are given in table 6 along with the comparisons on traditional steganography methods [23], [20], [22] and [7]. The first observation is that the hiding capacity of the method proposed is far higher than the traditional methods. Though the PSNR value of the proposed method is not very high in comparison with the statistical methods, hiding capacity of the proposed method is high compared to them. Also, the secret media used in the statistical methods are text and only [7] has used RGB image. There is always a trade-off between the performance and the hiding capacity. The proposed model has acceptable PSNR values with high hiding capacity.

Figure 17. Original cover and secret image with its histogram.

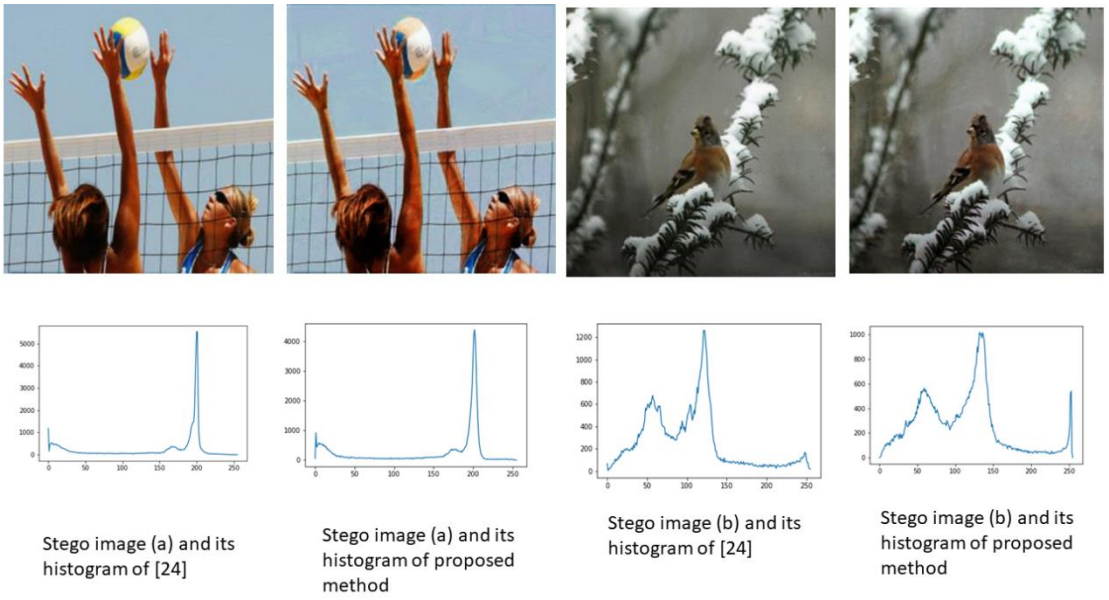

Figure 18. Stego image comparison between [24] and proposed method.

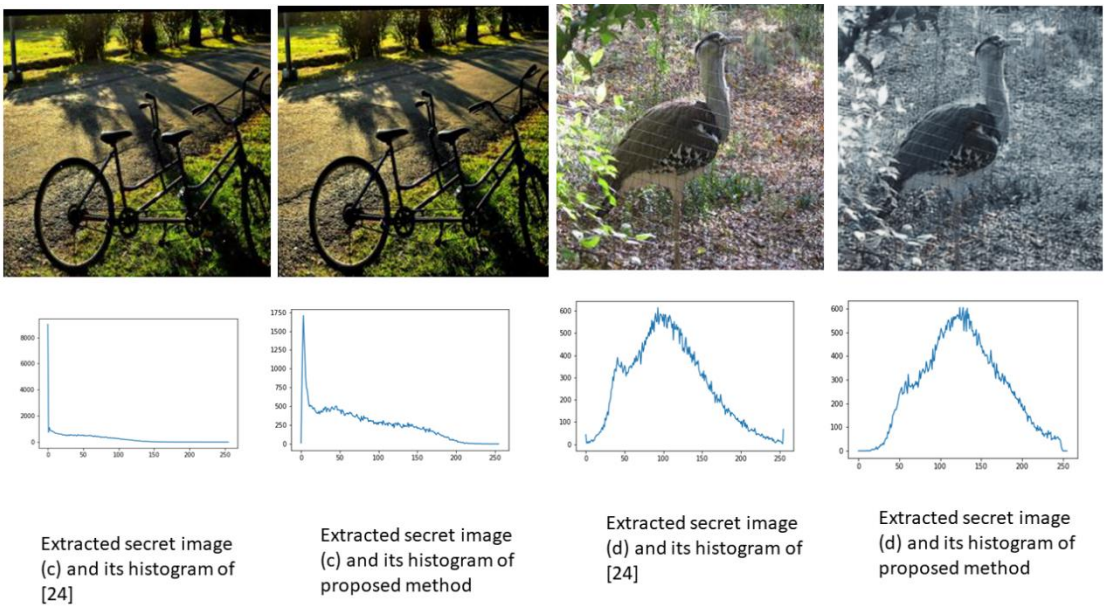

Figure 19. Extracted secret image comparison between [24] and proposed method

Table 6. Comparison of the Results Among Traditional Methods.

Method	Input	Secret media	Airplane	Baboon	Lena	Peppers
[23]	RGB	Text	37	37	37	37
[20]	Grayscale	Text	32.74	32.41	33.16	32.41
[7]	RGB	RGB	32.75	32.44	32.49	32.73
Proposed Method	RGB	RGB	33.70	30.21	31.49	32.14

The embedding network's PSNR value shows the security of the model proposed. PSNR value of the extraction network shows the robustness. Even though the secret image and the cover image are of the same dimensions, the embedding is not

visible. Another important point is that the image channels used are 3 (RGB) for both the input images. Nevertheless, visibility and distortion are minimum in the embedding network and the loss of recovered information in the secret image is also less.

In general, the proposed method has proved to perform the best among other methods in terms of capacity, security, perceptibility, and robustness. Not only it outperformed the traditional steganography methods, but it also outperformed other GAN-based and CNN-based methods.

Summary

A detailed report on the results along with the image results and a comparison between the proposed model and other previous works are given in this chapter. Comparison and evaluation of the proposed method with traditional and deep learning techniques are also given. An elaborate discussion on the observations made from the results obtained is also included in this chapter.

CHAPTER 6: CONCLUSION

A simple and lightweight deep convolutional autoencoder network is proposed for image steganography. The input image and the secret information used in the proposed method are RGB images. The proposed model consists of three parts – the preprocessing module, the embedding network, and the extraction network. The preprocessing module extracts the feature vectors from the input and the secret image concurrently. The feature vector is concatenated and fed to the embedding network to reconstruct an image that looks like the cover image in resemblance. Inversely, the extraction network extracts the embedded image from the stego image.

Training and testing are conducted on three datasets – COCO, CelebA, and ImageNet. PSNR and MSE are the evaluation metrics used. Along with the metrics, the image results are displayed to show the higher invisibility of the proposed method. The results from the proposed model are compared against traditional methods and other CNN-based and GAN-based methods. The proposed method has proved to outperform in terms of hiding capacity, security, robustness, and invisibility.

The PSNR value of the embedding network on the testing data which is not shown to the model during training is 31.96, 32.26, and 31.18 for COCO, CelebA, and ImageNet datasets respectively. Similarly, the PSNR value of the extraction module is approximately 27.9 for COCO, CelebA, and ImageNet datasets. The higher values of PSNR of the embedding network shows the security. The robustness of the extraction network is proved by the higher values of extraction PSNR.

6.1 Future Research Direction

Some of the gaps that can be found in the existing research which will be addressed in the future are briefed below,

- The proposed method can accept input images of fixed size (256 X 256). The model can be customized to accept images of arbitrary sizes.
- The model accepts two input images and hides the secret image inside the input image. The model architecture can be customized to accept the text as secret information as well. A single model can accept either text or image as secret information to produce a stego image.
- The proposed model is constructed completely using convolutional layers. Exploration of depth-wise separable convolutional layers can be done.
- Analysis of various attacks to determine the security of the proposed method against various attacks during network transfer.

REFERENCES

[1] L. Singh, A. K. Singh, and P. K. Singh, “Secure data hiding techniques: a survey,” *Multimed. Tools Appl.,* 2020, 79(23), pp.15901-15921 doi: 10.1007/s11042-018-6407-5.

[2] K. Pathak, S. Silakari, and N. S., “A Survey of Data Hiding Techniques,” *Int. J. Comput. Appl.,* 2018, 975, p.8887. doi: 10.5120/ijca2018918138.

[3] O. G. Abood and S. K. Guirguis, “A Survey on Cryptography Algorithms,” *Int. J. Sci. Res. Publ.,* 2018, 8(7), pp.410-415. doi: 10.29322/ijsrp.8.7.2018.p7978.

[4] C. Kumar, A. K. Singh, and P. Kumar, “A recent survey on image watermarking techniques and its application in e-governance,” *Multimed. Tools Appl.,* 2018, 7 7(3), pp.3597-3622. doi: 10.1007/s11042-017-5222-8.

[5] F. Boenisch, “A Survey on Model Watermarking Neural Networks,” no. Ml, 2020, [Online]. Available: http://arxiv.org/abs/2009.12153.

[6] J. Wang, M. Cheng, P. Wu, and B. Chen, “A Survey on Digital Image Steganography,” *J. Inf. Hiding Priv. Prot.,* 2019, 1(2), p.87. doi: 10.32604/jihpp.2019.07189.

[7] O. Elharrouss, N. Almaadeed, and S. Al-Maadeed, “An image steganography approach based on k-least significant bits (k-LSB),” *2020 IEEE Int. Conf. Informatics, IoT, Enabling Technol. ICIoT 2020,* no. May, pp. 131–135, 2020, doi: 10.1109/ICIoT48696.2020.9089566.

[8] S. Baluja, “Hiding images in plain sight: Deep steganography,” *Advances in Neural Information Processing Systems*., 2017, pp. 2069-2079.

[9] P. Techscholar, V. Prof, P. Kumar, and V. K. Sharma, “Information Security Based on Steganography & Cryptography Techniques : A Review,” *Int. J. Adv. Res. Comput. Sci. Softw. Eng.,* 2014, 410, pp. 246-250.

[10] M. S. Chandini, “An Overview about a Milestone in Information Security : STEGANOGRAPHY,” *International Journal of Progressive Research in Science and Engineering*, 2020, 1(2), pp.33-35.

[11] J. Xu., L. Xiang., Q. Liu., H. Gilmore., J. Wu., J. Tang., and A. Madabhushi., “Stacked sparse autoencoder (SSAE) for nuclei detection on breast cancer histopathology images,” *IEEE Trans. Med. Imaging*, 2016, 35(1), pp.119-130, doi: 10.1109/TMI.2015.2458702.

[12] D. Del Testa and M. Rossi, “Lightweight Lossy Compression of Biometric Patterns via Denoising Autoencoders,” *IEEE Signal Process. Lett.*, 2015, 22(12), pp.2304-2308, doi: 10.1109/LSP.2015.2476667.

[13] S. Anwar, M. Tahir, C. Li, A. Mian, F. S. Khan, and A. W. Muzaffar, “Image Colorization: A Survey and Dataset,” *arxiv*, pp. 1–17, 2020, [Online]. Available: http://arxiv.org/abs/2008.10774.

[14] K. Bajaj, D. K. Singh, and M. A. Ansari, “Autoencoders Based Deep Learner for Image Denoising,” *Procedia Comput. Sci.*, vol. 171, no. 2019, pp. 1535–1541, 2020, doi: 10.1016/j.procs.2020.04.164.

[15] V. V. Kuznetsov, V. A. Moskalenko, and N. Y. Zolotykh, “Electrocardiogram Generation and Feature Extraction Using a Variational Autoencoder,” *arxiv*, pp. 1–6, 2020, [Online]. Available: http://arxiv.org/abs/2002.00254.

[16] N. F. Johnson and S. Jajodia, “Exploring steganography: Seeing the unseen,” *Computer (Long. Beach. Calif).*, 1998, 31(2), pp.26-34, doi: 10.1109/MC.1998.4655281.

[17] S. Gupta, G. Gujral, and N. Aggarwal, “Enhanced Least Significant Bit algorithm For Image Steganography,” *IJCEM Int. J. Comput. Eng. Manag. ISSN*, 2012, 15(4), pp. 40-42.

[18] R. Das and T. Tuithung, “A novel steganography method for image based on Huffman Encoding,” *3rd National Conference on Emerging Trends and Applications in Computer Science*, 2012, pp. 14-18, doi: 10.1109/NCETACS.2012.6203290.

[19] B. Ida Seraphim, B. Sowmiya, and J. Anitha, “Enhanced image steganography scheme through multiway PVD,” *Int. J. Control Theory Appl.*, 2016.

[20] G. Swain, “Very High Capacity Image Steganography Technique Using Quotient Value Differencing and LSB Substitution,” *Arab. J. Sci. Eng.*, 2019, 44(4), pp.2995-3004, doi: 10.1007/s13369-018-3372-2.

[21] S. Wang, J. Sang, X. Song, and X. Niu, “Least significant qubit (LSQb) information hiding algorithm for quantum image,” *Meas. J. Int. Meas. Confed.*, 73, pp.352-359., 2015, doi: 10.1016/j.measurement.2015.05.038.

[22] N. Patel and S. Meena, “LSB based image steganography using dynamic key cryptography,” *International Conference on Emerging Trends in Communication Technologies (ETCT)*, 2017, doi: 10.1109/ETCT.2016.7882955.

[23] A. Qiu, X. Chen, X. Sun, S. Wang, and G. Wei, “Coverless Image Steganography Method Based on Feature Selection,” *J. Inf. Hiding Priv. Prot.*, 2019, 1(2), pp. 49, doi: 10.32604/jihpp.2019.05881.

[24] M. Hassaballah and A. I. Awad, *Deep Learning in Computer Vision: Principles and Applications.* CRC Press, no. March. 2020.

[25] I. Masi, Y. Wu, T. Hassner, and P. Natarajan, “Deep Face Recognition: A Survey,” *Proc. - 31st Conf. Graph. Patterns Images, SIBGRAPI 2018*, pp. 471–478, 2019, doi: 10.1109/SIBGRAPI.2018.00067.

[26] G. Ciaparrone, F. Luque Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, and F.

Herrera, “Deep learning in video multi-object tracking: A survey,” *Neurocomputing*, 2020, 381, pp.61-88, doi: 10.1016/j.neucom.2019.11.023.

[27] S. Herath, M. Harandi, and F. Porikli, “Going deeper into action recognition: A survey,” *Image Vis. Comput.*, 2017, 60, pp.4-21, doi: 10.1016/j.imavis.2017.01.010.

[28] R. Zhang, S. Dong, and J. Liu, “Invisible steganography via generative adversarial networks,” *Multimed. Tools Appl.*, 2019, 78(7), pp.8559-8575., doi: 10.1007/s11042-018-6951-z.

[29] P. Wu, Y. Yang, and X. Li, “StegNet: Mega Image steganography capacity with deep convolutional network,” *Futur. Internet*, 2018, 10(6), p.54., doi: 10.3390/FI10060054.

[30] P. Wu, Y. Yang, and X. Li, “Image-into-image steganography using deep convolutional network,” *Pacific Rim Conference on Multimedia*, 2018, pp. 792-802, doi: 10.1007/978-3-030-00767-6_73.

[31] A. ur Rehman, R. Rahim, S. Nadeem, and S. ul Hussain, “End-to-end trained CNN encoder-decoder networks for image steganography,” *Proceedings of the European Conference on Computer Vision (ECCV)*, 2019, doi: 10.1007/978-3-030-11018-5_64.

[32] X. Duan, K. Jia, B. Li, D. Guo, E. Zhang, and C. Qin, “Reversible image steganography scheme based on a U-net structure,” *IEEE Access*, 2019, 7, pp.9314-9323, doi: 10.1109/ACCESS.2019.2891247.

[33] T. Van Pham, T. H. Dinh, and T. M. Thanh, “Simultaneous convolutional neural network for highly efficient image steganography,” *Proc. - 2019 19th Int. Symp. Commun. Inf. Technol. Isc. 2019*, pp. 410–415, 2019, doi: 10.1109/ISCIT.2019.8905216.

[34] Z. Wang, N. Gao, X. Wang, J. Xiang, and G. Liu, *STNet: A Style Transformation Network for Deep Image Steganography*, vol. 11954 LNCS. Springer International Publishing, 2019.

[35] K. Yang, K. Chen, W. Zhang, and N. Yu, *Provably secure generative steganography based on autoregressive model*, vol. 11378 LNCS. Springer International Publishing, 2019.

[36] I. J. Goodfellow et al., "Generative adversarial nets," *Advances in neural information processing systems*, 2014, pp. 2672-2680.

[37] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," *Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia*, 2017.

[38] X. Huang, Y. Li, O. Poursaeed, J. Hopcroft, and S. Belongie, "Stacked generative adversarial networks," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5077-5086, doi: 10.1109/CVPR.2017.202.

[39] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," *Proceedings of the IEEE international conference on computer vision*, 2017, doi: 10.1109/ICCV.2017.244.

[40] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," *arxiv*, pp. 1–7, 2014, [Online]. Available: http://arxiv.org/abs/1411.1784.

[41] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," *International conference on machine learning*, 2017.

[42] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing

generative adversarial nets," *Advances in neural information processing systems*, 2016, pp. 2172-2180.

[43] D. J. Im, C. D. Kim, H. Jiang, and R. Memisevic, "Generating images with recurrent adversarial networks," *arxiv*, 2016, [Online]. Available: http://arxiv.org/abs/1602.05110.

[44] D. Volkhonskiy, I. Nazarov, and E. Burnaev, "Steganographic generative adversarial networks," *Twelfth International Conference on Machine Vision (ICMV 2019)*, 2019, 11433, p. 114333M, doi: 10.1117/12.2559429.

[45] H. Shi, J. Dong, W. Wang, Y. Qian, and X. Zhang, "SSGAN: Secure steganography based on generative adversarial networks," *Pacific Rim Conference on Multimedia*, 2018, pp. 534-544, doi: 10.1007/978-3-319-77380-3_51.

[46] H. Shi, X. Y. Zhang, S. Wang, G. Fu, and J. Tang, "Synchronized Detection and Recovery of Steganographic Messages with Adversarial Learning," *International Conference on Computational Science*, 2019, doi: 10.1007/978-3-030-22741-8_3.

[47] R. Meng, Q. Cui, Z. Zhou, Z. Fu, and X. Sun, "A Steganography Algorithm Based on CycleGAN for Covert Communication in the Internet of Things," *IEEE Access*, 2019, 7, pp.90574-90584, doi: 10.1109/ACCESS.2019.2920956.

[48] C. Chu, A. Zhmoginov, and M. Sandler, "CycleGAN, a Master of Steganography," *arxiv*, pp. 1–6, 2017, [Online]. Available: http://arxiv.org/abs/1712.02950.

[49] P. G. Kuppusamy, K. C. Ramya, S. Sheeba Rani, M. Sivaram, and V. Dhasarathan, "A novel approach based on modified cycle generative adversarial networks for image steganography," *Scalable Comput.*, 2020, 21(1), pp.63-72,

doi: 10.12694/SCPE.V21I1.1613.

[50] H. Porav, V. Musat, and P. Newman, “Reducing Steganography In Cycle-consistency GANs,” *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Work.*, 2019.

[51] J. Yang, D. Ruan, J. Huang, X. Kang, and Y. Q. Shi, “An Embedding Cost Learning Framework Using GAN,” *IEEE Trans. Inf. Forensics Secur.*, 2020, 15, pp.839-851, doi: 10.1109/TIFS.2019.2922229.

[52] W. Tang, S. Tan, B. Li, and J. Huang, “Automatic Steganographic Distortion Learning Using a Generative Adversarial Network,” *IEEE Signal Process. Lett.*, 2017, 24(10), pp.1547-1551, doi: 10.1109/LSP.2017.2745572.

[53] M. M. Liu, M. Q. Zhang, J. Liu, P. X. Gao, and Y. N. Zhang, “Coverless Information Hiding Based on Generative Adversarial Networks,” *Yingyong Kexue Xuebao/Journal Appl. Sci.*, 2018, doi: 10.3969/j.issn.0255-8297.2018.02.015.

[54] Z. Zhang, G. Fu, J. Liu, and W. Fu, *Generative information hiding method based on adversarial networks*, vol. 905. Springer International Publishing, 2020.

[55] B. Chen, J. Wang, Y. Chen, Z. Jin, H. J. Shim, and Y. Q. Shi, “High-capacity robust image steganography via adversarial network,” *KSII Trans. Internet Inf. Syst.*, 2020, 14(1), doi: 10.3837/tiis.2020.01.020.

[56] K. A. Zhang, A. Cuesta-infante, L. Xu, K. Veeramachaneni, and J. Carlos, “SteganoGAN : High Capacity Image Steganography with GANs,” *arxiv*, 2018.

[57] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, “HiDDeN: Hiding data with deep networks,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11219 LNCS, pp. 682–697, 2018, doi: 10.1007/978-3-030-01267-0_40.

[58] X. Duan, H. Song, C. Qin, and M. K. Khan, “Coverless steganography for digital images based on a generative model,” *Comput. Mater. Contin.*, 2018, 55(3), pp.483-493, doi: 10.3970/cmc.2018.01798.

[59] X. Duan, B. Li, D. Guo, Z. Zhang, and Y. Ma, “A coverless steganography method based on generative adversarial network,” *Eurasip J. Image Video Process.*, 2020, pp.1-10, doi: 10.1186/s13640-020-00506-6.

[60] Z. Wang, N. Gao, X. Wang, X. Qu, and L. Li, “SSteGAN: Self-learning steganography based on generative adversarial networks,” *International Conference on Neural Information Processing*, 2018, doi: 10.1007/978-3-030-04179-3_22.

[61] J. Hayes and G. Danezis, “Generating steganographic images via adversarial training,” *Advances in Neural Information Processing Systems*, 2017, 30, pp.1954-1963.

[62] P. Bas, T. Filler, and T. Pevný, “‘Break our steganographic system’: The ins and outs of organizing BOSS,” *International workshop on information hiding*, 2011, pp 59-70, doi: 10.1007/978-3-642-24178-9_5.

[63] Jia Deng, Wei Dong, R. Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, “ImageNet: A large-scale hierarchical image database,” 2009, doi: 10.1109/cvprw.2009.5206848.

[64] T. Y. Lin *et al.*, “Microsoft COCO: Common objects in context,” 2014, doi: 10.1007/978-3-319-10602-1_48.

[65] Z. Liu, P. Luo, X. Wang, and X. Tang, “Deep learning face attributes in the wild,” 2015, doi: 10.1109/ICCV.2015.425.