e-ISSN:2582-7219

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

Volume 7, Issue 5, May 2024

INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 7.521

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| Monthly Peer Reviewed & Referred Journal |

| Volume 7, Issue 5, May 2024 |

| DOI:10.15680/IJMRSET.2024.0705098 |

Multi-Image Concealment in High- Resolution Carriers Using Deep Neural Networks

Vaishnavi S, Kavitha R

Department of Computer Science and IT, JAIN (Deemed-to-be University), Bangalore, India

Department of Computer Science and IT, JAIN (Deemed-to-be University), Bangalore, India

ABSTRACT: The discipline of concealing a private message within an ordinary public message is referred to as steganography. Over time, steganography has utilized techniques such as LSB manipulation and other simple methods to embed images of lower quality into those of higher quality. Our objective is to employ deep neural networks in the process of concealing and revealing multiple concealed images within a single high-resolution cover image. Deep neural networks are designed to operate in tandem, undergoing simultaneous training to enable both the hiding and revealing procedures. Following training on randomly chosen images from the ImageNet database, the system demonstrates effective performancewith authentic photographs from a diverse array of origins.

KEYWORDS Data security, Neural networks, Deep Learning, ImageNet dataset, Information hiding, Convolutional NeuralNetworks (CNNs)

I. INTRODUCTION

In the modern world, machine learning (ML) has emerged as a keystone of innovation, drastically transforming sectors and altering the way we approach issues. ML algorithms, which provide computers the ability to learn from data, automate processes, forecast outcomes, and reveal insights at speeds and sizes that beyond human capabilities. Automation boosts innovation in a variety of industries, including healthcare, banking, cybersecurity, and transportation, along with the increasing efficiency.

Additionally, ML Across industries and applications, including machine learning and data-driven decision-making, traditional data transmission methods continue to be crucial for facilitating effective, dependable, and secure communicationbetween devices and systems. The technique known as steganography, which covers secret messages in obviously normal data, has several benefits across a range of industries. It allows invisible delivery of sensitive data without causing any fear by wrapping it in simple files. Steganography protects secret by invisibility, making it more difficult for unauthorised partiesto discover hidden communications, in opposed to encryption, which may indicate the presence of sensitive material. Steganography is particularly useful for putting watermarks or copyright information in videos since it makes data hiding possible without noticeably changing the appearance of the cover object. Because crucial information is hidden inside seemingly innocent-looking files, this technique also adds a higher level of security against data theft. Steganography additionally aids in the process of hiding confidential information in digital images, allowing an unseen and invisible way ofcommunicating. In order to include the hidden data, this technique modifies the cover image's metadata or pixel values in anundetectable way.

Utilising machine learning (ML) algorithms, steganography preserves the integrity and appearance of the cover item while hiding sensitive information within data. With the least number of detectable alterations, ML-based steganography approaches are aimed at integrating secret messages into several kinds of digital data, including text documents, audio files,and photographs.

One commonly employed technique is the methodology known as Least Significant Bit (LSB) substitution, where the least significant bits of pixel values are interchanged with bits from the secret message. This strategy ensures that any alterations remain imperceptible to human vision, thus preserving the integrity of the image while embedding confidential information.

And also in this regard, recurrent neural networks (RNNs) can be trained to conceal messages inside text data by creating sequences that encode the secret message, while convolutional neural networks (CNNs) can be trained to discreetly alter pixel values in images to encode hidden information. Utilising generative adversarial networks (GANs) to manipulate the generator network in order to produce data that looks realistic and contains the secret message is another method for embedding hidden information within digital media. Advantages of ML-based steganography such as improved efficiency and safety since algorithms can dynamically learn to hide communications while reducing noticeable changes. even with its applications in data security, privacy protection, and covert communication, steganography in machine learning continues to be a vibrant and developing discipline that presents exciting opportunities for research and development.

Deep neural networks (DNNs) were recently encompassed into data-hiding pipelines, which is an important development that improves the quality and privacy of encoded information. Conventional methods tend to find it difficult to successfully hide information while maintaining the carrier signal's integrity. On the other hand, DNN-based methods have become a game-changing fix. DNNs in data-hiding pipelines represent a big step forward in information security, providing unbeatable secrecy and quality in encoded messages. As research continues, deep learning-driven innovations are expected to redefine the bounds of concealing and protecting sensitive information.

CHALLENGES: Effective steganography poses challenges due to the impact of embedding a message on the appearance and content of the carrier. The degree of alteration is influenced by two key factors: the volume of data to be concealed, withtextual content sometimes concealed within visuals, and the amount of data hidden, typically measured in bits-per-pixel (bpp) at 0.4bpp or lower. Larger messages correspond to higher bpp values, resulting in more significant changes to the carrier. The extent of modification is contingent upon the characteristics of the carrier image, with concealing information in noisy, high- frequency areas of an image presenting fewer discernible hurdles compared to concealing it in uniform regions.

OBJECTIVE OF THIS PAPER: Drawing upon the concepts introduced in previous works such as "Hide and Speak" and "Deep Steganography for Hiding Images in Plain Sight: In the Direction of Deep Neural Networks for Speech," we aim to adopt a similar approach by utilizing these two publications to embed multiple images within a single cover image. Our methodology diverges from conventional techniques as we amalgamate cover and hidden images of the same resolution, with the objective of ensuring that any modifications made to the encoded cover image are undetectable through both statistical analysis and human observation.

II. LITERATURE REVIEW

The two implementations listed below are the most significant and in line with our objective out of all of them.
(Baluja) The objective of this study is to embed a full-scale colored image within a smaller one. Deep neural networks, configured to function as a collective, undergo training to execute both the hiding and revealing procedures simultaneously. Utilizing randomly chosen images from the ImageNet database for training, the system demonstrates impressive performance when handling naturally captured images from diverse origins. Differing from prevalent steganographic methods that hide confidential data in the less significant regions of the cover image, their approach entails compressing and distributing the encoding of the concealed image across all available bits.

The three aspects of the system that are involved are

1. **Preparation Network:** The hidden image must be established by the Preparation Network. When the dimensions of the hidden picture (dimensions MM) are less than those of the cover image (N N), it gradually modifies the hidden image's sizeto match the cover images. This procedure assures that the components of the hidden image are distributed over all N N pixels.

2. **Hiding Network:** The cover image and the output from the preparation network are used by the Hiding Network to create the Container image. The RGB channels of the cover picture concatenated based on depth and the modified components of the concealed image make up the N N pixel array that is fed into the network.

3. **Reveal Network:** The image receiver makes use of this decoder. It just gets the container's image; neither the cover imagenor the concealed image are sent to it. The decoder network removes the cover image, revealing the secret image.

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| Monthly Peer Reviewed & Referred Journal |

| Volume 7, Issue 5, May 2024 |

| DOI:10.15680/IJMRSET.2024.0705098 |

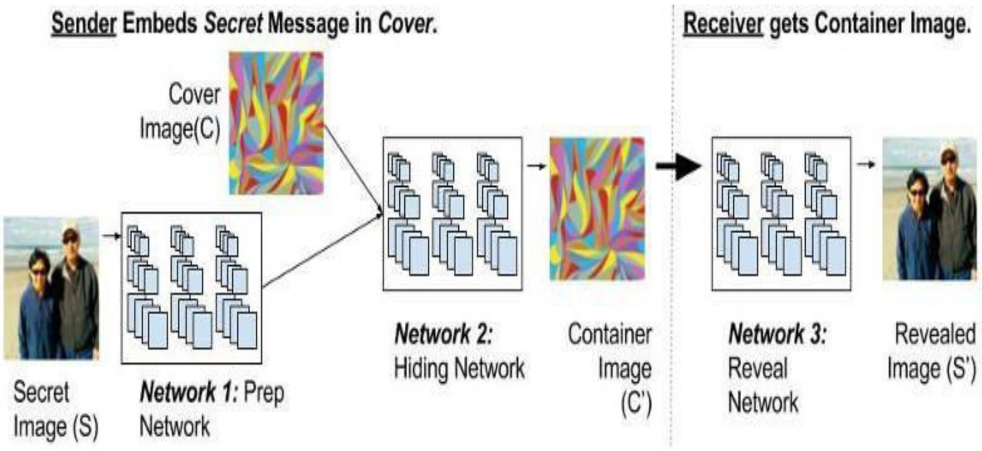
Sender Embeds Secret Message in Cover.
Receiver gets Container Image.
Cover Image(C)
Secret Image (S)
Network 1: Prep Network
Network 2: Hiding Network
Container Image (C')
Network 3: Reveal Network
Revealed Image (S')

Figure 1. The three elements that make up the entire system. Left: Setting up the covert image. Centre: Keeping thepicture hidden within the cover photo. Right in that Using the reveal network to uncover the secret picture; this is trained simultaneously and utilised by the person who receives it.

In the study, (Baluja 2017) However, there has never been an attempt to consciously conceal the presence of that knowledgefrom computer detection.

They trained the steganalysis networks as binary classifiers using the intact ImageNet images as negative samples and their containers as positive scenarios. The study offers a baseline for a single secret image's encoding. However, multi-image steganography is not covered by it.

(Felix Kreuk, 2019) Here to implement steganography for speech data. Its foundation is an architecture made up of three subsystems: a message decoder network, a carrier decoder network, and an encoder network. They use concepts from pervious papers to enable audio signals to be encoded by the encoder network.

The model's architecture is divided into three smaller components:

- An Encoder Network (Ec)
- ACarrier Decoder Network (Dc)
- AMessage Decoder Network (Dm)

After receiving this output, the Carrier Decoder (Dc) generates the carrier, which contains a concealed message. Finally, information is received by the Message Decoder (Dm), which then reconstructs the hidden message.

How to map potential redundancy using the carrier signal is covered in the first part. The second phase uses the map to "stuff"a secret message into the carrier with the least possible influence on it. The last section describes how to interpret the steganographically altered hidden message from the carrier. Gated convoys make up the entirety of these networks. Three blocks of gated convoys, Dc four and Dm six blocks, make up Ec. There are 64 3*3 kernels in each block.

This study demonstrates how to hide many hidden messages within a single carrier, which is in line with our goals. The research uses a single speech recording to encrypt five different voice messages. Two different approaches are used to achieve this. Several decoders, each trained to interpret a different message, are used in one way. The other approach uses a conditionaldecoder that accepts as input as well.

We extracted many secret images from the cover image by applying the concept of multiple decoders, which we borrowed from this paper. Although the cover image seems to be the secret image, the secret images are actually concealed inside it and are not visible until they pass through several preparatory networks. We introduced multi-image steganography by extending the idea from this paper to images, allowing many images to be included in a single cover image and subsequently retrieved.

III. PROPOSED METHODOLOGY

The goal of this paper is to conceal more than two pictures behind only one front image. It is required to retrieve the embedded hidden photos with the least degree of loss. The original cover image should be similar to the encoded version.
We use the concept of (Baluja) and (Felix Kreuk, 2019) to do this. We use the idea of network implementation from (Baluja) in which a prep and hidden network acts as an encoder and a reveal network as a decoder.

To expand this for many images, we first concatenate the resultant data with the carrier image using the prep network, and then send the combined data over the hidden network. Next, we apply the idea of having multiple decoders, one for each secret image, as suggested by (Felix Kreuk, 2019)to extract all of the secret images from the container image.

Rather than placing the secret images at the LSBs of the original cover image, we build on the idea from (Baluja) to improve the securityof our image retrieval algorithm by introducing secret images with noise into the original image. (Felix Kreuk, 2019) implementing multiple audio signal steganography utilising several decoders.

The visual domain also makes use of this method. It is not necessary to decrease the size of the image or jeopardise the hidden images' colour pathways. An alternative method substitutes a single conditional decoder for several decoders. Nevertheless, we have limited our efforts to developing numerous pre/reveal networks in this iteration. We therefore made the decision tobase our extension on this methodology.

The encoder/decoder architecture using this method can be briefly described as follows:

- **ENCODER:** is made up of several prep networks, each of which corresponds to a different secret image input. PrepareNetwork outputs are transmitted via the Hiding network after being concatenated with the cover image.
- **DECODER:** The decoder network consists of many reveal networks, each of which has undergone independent trainingfor decoding the message with which it correlates.

SYSTEM ARCHITECTURE

This explains all the network layers in the deep steganography as shown in figure 2. Each of the separate networks has thefollowing underlying architecture.

- Prep is the first type of network. **Prep Networks**: Each prep network is made up of two layers. made up of three different Conv2D layers for every layer. The channels for each of these three Conv2D layers are fifty, ten and five respectively, and the fundamental kernel sizes for each layer are three, four, and five On both axes, the path length never deviates from one. The right amount of padding is added to every Conv2D layer in order to preserve the dimensions of the final image. Following each Conv2d layer is a ReLUactivation.
- The hidden network is the second network. **Hiding Network:** The network of hidden levels consists of five tiers. Each of these layers is composed of the three unique Conv2D layers. There is a comparable underlying structure between the Conv2D levels in the Prep Network and the Conv2D layers in the Hidden Network.
- Third network is reveal network. **Reveal Network:** Using five layers of similarly constructed Conv2D layers, each reveal network has an underlying design that is comparable to that of the hiding network.

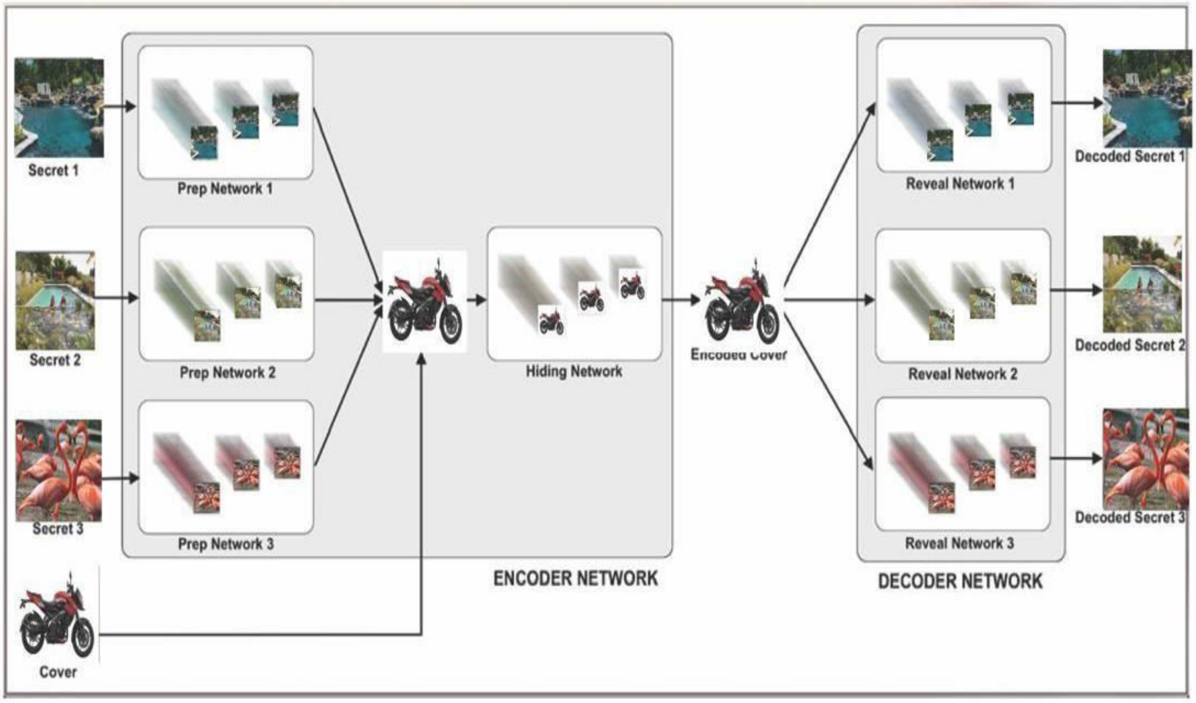

Figure 2. Multi steganography using deep neural network

IV. EXPERIMENTAL ANALYSIS

Since our model has no unique constraints for the picture classes, we used the Tiny ImageNet (tin) dataset from Kaggle to obtain the cover and secret photos.

The Stanford CS231 class uses the 64 *64* 3 image collection that makes up the dataset. Larger photos from datasets like ImageNet can also be used with further modifications to the final model (Deng et al., 2009). For easier training, Tiny ImageNet has also been utilised.

EXTRACT IMAGES FROM DATASET: This step involves retrieving images from a dataset to be used for training the autoencoder.

MAKE RANDOM PAIR OF COVER AND SECRET IMAGES: In this step, a random pair of images is created, where one image (the "cover") is used as the input to the autoencoder, and the other image (the "secret") is used as the target output. **INPUT IMAGE TO ENCODER:** The cover picture is then fed into the encoder network, which starts the process of compressing it into a lower-dimensional representation.

IMAGE PROCESSED BY PREPAREATION NETWORK: The image is processed by a preparation network, which mayinclude steps such as resizing, normalization, or other preprocessing techniques.

IMAGE HIDDEN BY HIDDEN NETWORK (ENCODED COVER): The prepared image is then passed through the hidden layers of the encoder network, which compresses the image into a lower-dimensional representation or "encoding".

IMAGE DECODED BY DECODER NETWORK: The encoded image is subsequently routed to the decoder network, which attempts to reconstruct the original image from the encoded representation.

ERROR IS CALCULATED AND WEIGHTS ARE IMPROVED An error metric is calculated by subtracting the originalsecret image from the decoded image. This error is then used to update the weights of the autoencoder network, improving its ability to encode and decode images.

OUTPUT IS SHOWN AFTER SOME EPOCHS: After several iterations or "epochs" of training, the autoencoder's outputis shown to evaluate its performance.

IF NUMBER OF EPOCHS REACHED TRAINING COMPLETE: If a certain amount of epochs is achieved, the trainingprocess is considered finished.

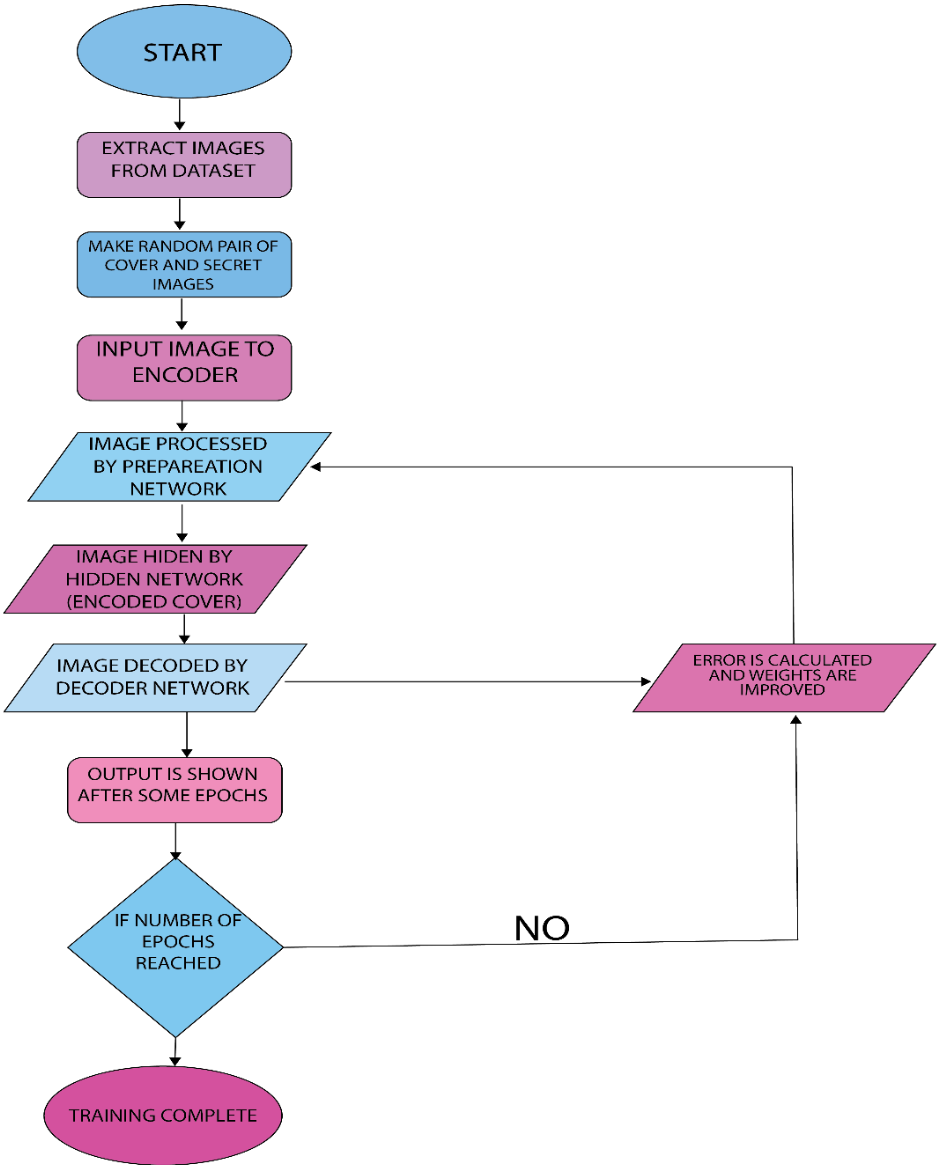
START
EXTRACT IMAGES FROM DATASET
MAKE RANDOM PAIR OF COVER AND SECRET IMAGES
INPUT IMAGE TO ENCODER
IMAGE PROCESSED BY PREPAREATION NETWORK
IMAGE HIDEN BY HIDDEN NETWORK (ENCODED COVER)
IMAGE DECODED BY DECODER NETWORK
ERROR IS CALCULATED AND WEIGHTS ARE IMPROVED
OUTPUT IS SHOWN AFTER SOME EPOCHS
IF NUMBER OF EPOCHS REACHED
NO
TRAINING COMPLETE

Figure 3. Flow of proposed model

Our training set consists of a random selection of photographs from each of the 200 categories. 2000 photographs are selectedat random. The image vectors are altered based on RGB values. The training data was separated into four parts: one half wasused for the front image, while the other three were utilised for three secret images.

An LR scheduler tailored for the Adam optimizer was used. The model was trained over 750 epochs with a total number of trials of 256, followed by 400 epochs with a batch size of 32. The 64x64 pictures from the Tiny Image Dataset have been used. The dataset is formed by training with 10 photos per class and testing with a total of 2000 images. The train set consistsof two components. The first 1000 shots serve as cover images, with the remaining 1000 utilised for training.

The stacked Keras model and loss is the same for preparation and hiding networks. Each reveal network has its own loss function and layered model. Both the cover image and the secret image loss are taken into account during the whole model training.

RESULTS

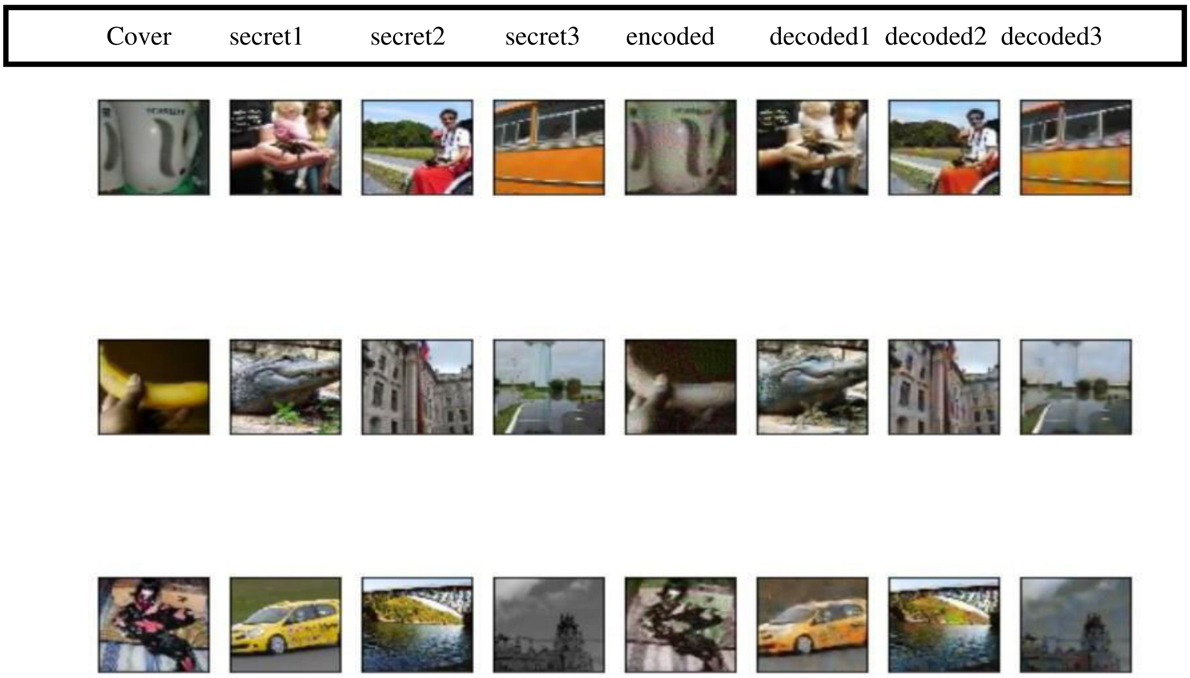

Figure 4. The result of concealing three concealed photos. Columns, starting from the right: Hidden Image, HiddenImage1, Hidden Image2, Hidden Image3, Encapsulated Wrap Image, Unveiled Hidden Images 1, 2, and 3.

Figure 4 shows the results of concealing three hidden images. The encoded cover is noisier than the scenario where only two hidden photographs are applied. It is projected that as the number of photos grows, more image features will be buried in a single image, increasing the loss of all values. Thus, we must establish a limit on the total number of photos that can be posted.

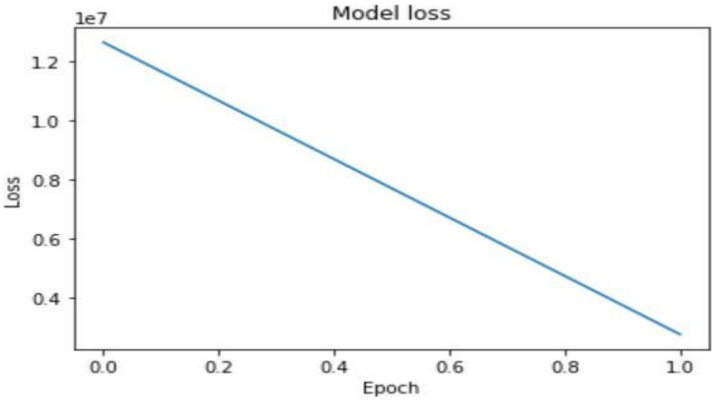

Figure 5. Model Loss

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| Monthly Peer Reviewed & Referred Journal |

| Volume 7, Issue 5, May 2024 |

| DOI:10.15680/IJMRSET.2024.0705098 |

The procedure given is for image-based autoencoder training. It entails pulling photographs from a collection and generatinga random set of cover and secret images. The cover picture is fed into the encoder network, which reduces it to a lower- dimensional representation. The encoded image is subsequently sent via the decoder network, which reconstructs the original image. The difference between the original and reconstructed images is utilised to adjust the autoencoder's weights, enhancing its capacity to encode and decode images. After multiple iterations or "epochs" of training, the autoencoder's output is evaluated. It should be emphasised that the Preparation and Hiding Networks, as well as the Reveal Network, have separate loss functions. We create layered Keras models, one for the Reveal Network.

POSSIBLE APPROACHES

From the standpoint of implementation, our goals are as follows:

- Maximize the number of secret photographs while minimising loss.
- Investigating s and c's impact on results.
- Rather than using several decoders, use conditional decoders

V. CONCLUSION

We developed a knowledge of image-based steganography. We were able to learn more by reading a variety of articles, andthe problem statement is crucial to data security. Our execution expanded the single image steganography model by putting into practice several reveal networks that correlate to Every concealed image, as proposed by While maintaining a respectable minimum loss for secret photos .We were able to encode and decode up to three different secret images using a single cover image of comparable size. However, the loss of our cover image was bigger. Instead of experimenting with other loss types that would have been more applicable for our model, we focused mostly on visual perception to assess overall loss.

REFERENCES

1. Tiny ImageNet Visual Recognition Challenge.
2. Baluja, S. Hiding images in plain sight: Deep steganogra phy. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H.,Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), Advances in Neural Information Processing Systems 30.
3. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei Fei, L. ImageNet: A Large-Scale Hierarchical Image Database.In CVPR09, 2009.
4. Hide and Speak: Towards Deep Neural Networks for Speech Steganography Felix Kreuk1, Yossi Adi2, Bhiksha Raj3,
5. Rita Singh3, Joseph Keshet1
6. Hiding Images in Plain Sight: Deep Steganography Shumeet Baluja Google Research Google, Inc. shumeet@google.com
7. Hayes, J. and Danezis, G. Generating steganographic im ages via adversarial training. In NIPS, 2017.
8. Ingham, F. Deepsteg: Implementation of hidding images in plain sight: Deep steganography in pytorch.
9. Kreuk, F., Adi, Y., Raj, B., Singh, R., and Keshet, J. Hide and speak: Deep neural networks for speech steganogra phy, 2019. Muzio, A.
10. Deep-steg: Implementation of hidding images in plain sight: Deep steganography in keras.
11. Zhu, J., Kaplan, R., Johnson, J., and Fei-Fei, L. Hidden: Hiding data with deep networks. CoRR, abs/1807.09937, 2018.
12. URL http://arxiv.org/ abs/1807.09937
13. Benedikt Boehm. Stegexpose- A tool for detecting LSB steganography. CoRR, abs/1410.6656, 2014.
14. Stegexpose- github. https://github.com/b3dk7/StegExpose.

INTERNATIONAL JOURNAL OF
MULTIDISCIPLINARY RESEARCH
IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com