Steganography Information Retrieval Mechanism Using Deep Neural Network

Hasan Kareem Abdulrahman
Computer Systems Department,
Northern Technical University,
Iraq
Hasan.abdulrahman@ntu.edu.iq

***Abstract*- Steganography has captivated the interest of a rising number of academics in recent years, as its applications have grown beyond information security. The most common approach is digital signal processing (DSP), which includes least significant bit (LSB) encoding. Deep learning has lately been used in various innovative approaches to the steganography problem. The bulk of existing methods, on the other hand, are designed for image in picture steganography. This study proposes using deep learning algorithms to disguise clandestine audio in digital photos. The first network conceals the concealed audio in a picture, while the second network decodes the image to retrieve the actual audio. In-depth research make use of a set of 24K images and the VIVOS Corpus audio dataset1. Experimental data shows that our strategy is more effective than earlier methods. Both the visual and audio integrity are preserved, and the maximum length of the concealed audio is substantially extended.**

Keywords: Image , FFNN, FFT, WLT, RGB, Steganography.

I. INTRODUCTION

Computers and the internet play an important role in many aspects of modern life, especially in information technology. There has been a rise in concerns about information security as the internet has grown in popularity. The two most fundamental solutions to this challenge are cryptography and steganography. Unlike cryptography, which aims to protect the substance of the communication, steganography entails concealing secret information inside conventional data types. It's usually used to communicate sensitive material to those who are already aware of its existence, while others regard it like ordinary data.

Hundreds of millions of photos and audio files are submitted to the internet every day. As a result, steganography is growing in popularity and being used more regularly. It's also used as a watermarking technique on photographs, music, or digital software for copyright protection, impersonation detection, duplication prevention, content validation, and monitoring or tracing of illegal copies, as well as advertising monitoring, in the entertainment and software industries.

In this paper, we look into audio-into-image steganography, which aims to hide hidden audio in digital picture data. Our approach is more difficult than concealing an image in an image or hiding audio in an audio since audio and pictures are in distinct domains. Audio data is a one-dimensional matrix with values ranging from 2 15 to 2 15 1 that depicts a series of amplitudes in the time domain. Image, on the other hand, is a three-dimensional matrix with values ranging from 0 to 255 that depicts the intensity of light. Image with 8 bits. Directly encoding the secret audio in the image's least significant bits is one of the most used approaches. This method, however, has been demonstrated to be worthless when the number of bits to hide is more than the number of bits that may be updated in the cover data. If more bits are altered, the integrity of the cover data is no longer protected. This is the main stumbling point in hiding audio data within a picture.

We introduce a Deep Convolutional Neural Network (DCNN) model capable of both hiding and revealing audio within an image in light of these challenges. The procedure of the model in question is depicted in Figure 1. The secret audio is pre-processed and buried in the cover picture to generate the container image. The container picture is then used to obtain the original concealed audio. Using our strategy, the difference between cover-container picture and secret-revealed audio pairs is essentially undetectable.

To summarise, our main contributions are: (1) suggesting the use of DNN to address the problem of secret audio being hidden in digital photographs; and (2) proposing the use of DNN to manage the problem of secret audio being hidden in digital images. (2) Comparing and contrasting the audio and visual quality of various audio data preprocessing techniques. (3) Demonstrating that our technique can conceal longer audio than other methods and that the difference between the cover and container images is difficult to discern with human eyes.

The rest of the paper is organised as follows. Section 2 provides a quick overview of key research. Section 3 will go through the suggested model architecture and data pre-processing strategy. Section 4 covers data preparation and system setup for the experiment. The results of the trials, as well as their appraisal, are provided in Section 5. The paper comes to an end with Section 6.

II. WORK IN CONNECTION

A. Previously employed steganography approaches

The history of steganography is lengthy and famous.

[2], [13] give in-depth descriptions of steganography's techniques and uses. According to ancient steganography methods, the secret message is physically buried inside another data type. These processes are easily observable due to their simplicity. As a result, stronger steganography is required. As a result of advancements in digital signal processing, many digital steganography techniques have been developed. The initial technique was presented by Kurak and McHugh [12], who successfully embedded data into four least significant bits (LSB) of a picture. A hidden message can also be buried in a data type other than an

This work is licensed under a Creative Commons Attribution-Noncommercial 4.0 International License

image. In [7], C. Hernandez et al. succeeded in concealing data in HTML and XML pages, as well as executable files. (.exe). In [8, Chet Hosmer used the LSB method to conceal data in GIF and JPEG format pictures, as well as audio files.

B. Deep learning and steganography

Although LSB-based steganography is widely utilised, it has a number of limitations. The LSB-based technique's primary flaw is that it is susceptible to steganalysis. Deep neural network-based approaches have been developed to solve the LSB technique's current difficulty. One of the first examples is Imran Khan's [10], which shows how feature representations may be taught automatically by a network with several convolutional layers.

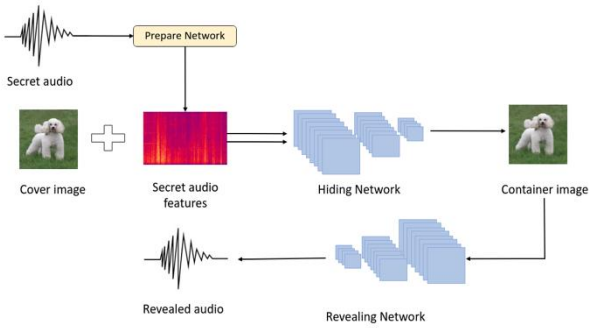

Fig. 1. The system general architecture.

In 2017, Shumeet Baluja utilised a three-convolutional neural network architecture to conceal a concealed picture within a public image of the same size [1]. Recently, Jiren Zhu et al. proposed an end-to-end deep neural network for both encoding and decoding a hidden message as a bit string within a picture [18].

Jiren Zhu proposed the adversarial network in addition to the encoder-decoder pair. The adversarial network can predict whether or not a given image includes a hidden message and can help improve the quality of the encoded image.

Steganography that encodes audio into an image is known as audio-into-image steganography.

Audio steganography is a subset of the larger topic of steganography that attempts to conceal secret messages inside audio recordings. A great deal of effort has gone into resolving this problem. [3]–[5] are some examples of representative literary works. When it comes to encoding and decoding audio files, a typical method is to utilise the Short Time Fourier Transform (STFT) to extract the spectrogram of the secret and carrier audio, and then use a pair of deep neural networks to encode and decode the audio. As explained in [17], Denpan Ye et al. use a generative adversarial network-based architecture with three parts: encoder, decoder, and steganalyzer. Denpan Ye's approach can not only encode and decode audio data, but it can also determine whether an audio file includes a secret message. Only a few attempts at audio-into-picture steganography have been made since audio and image data are in separate domains.

The most current research may be found in [9], [15]. The wavelet transform is used in both works to compress voice data and the LSB technique is used as an embedding approach. These approaches were presented many years ago, and little progress has been made since then. We are motivated by the lack of a good audio-into-image steganography technology. The STFT approach is employed for audio processing, while a deep convolutional neural network handles the encoding and decoding stages. Our solution can take use of the learning skills of deep networks as well as the information extraction capabilities of digital signal processing techniques by merging the two.

III. PROPOSED METHOD

The procedure for normalising images is simple: the value of each pixel is divided by 255 to generate a new matrix in the value domain ranging from 0 to 1. For audio data, however, there is a somewhat different processing technique. The amplitude of audio data in 16-bit PCM format varies from 2 15 to 2 15 1.

As a result, traditional methods such as min-max scaling and mean-standard deviation normalisation are unsuccessful. As a result, various studies concentrating on analysing audio histograms are conducted in order to establish the optimal normalisation method.

Pre-processing audio data may be done in two ways.

- Method 1: Raw audio data is normalised and moulded into a new three-dimensional matrix with the same form as the picture with three colour channels.
- Method 2: The short-time Fourier transform (STFT) is employed to translate audio to the frequency domain. The STFT of a windowed signal is a set of Fourier transforms.

The STFT provides time-localized frequency information when the frequency components of a signal vary over time. The translated data is now a three-dimensional matrix with two channels [6], [14], rather than three channels.

A. Structure of the model

As previously described, we develop a DNN model, especially a convolutional neural network (CNN), that is capable of concealing audio in pictures while revealing the real audio. The preparing network and the concealing network are the first two sub-models, while the reveal network is the second:

- For encoding, there is a submodel:
    - Prepare network: accurately extracts the attributes of the audio before concatenating it with the picture.
    - Hiding network: creates the container image by masking the cover picture's secret audio properties.
- Reveal network: extracts the original audio by decoding the container image. • Submodel for decoding:

Our strategy's overall design is similar to that of Baluja et al. [1]. The entire architecture is seen in Figure 1.

However, each component of our proposed convolutional neural network has been adjusted to make it more suited for audio data while also reducing training time. The entire architecture differs due to variances in audio pre-processing techniques. Several adjustments are necessary to make the overall design more suitable for each preprocessing method.

The architecture used by all three network components is known as the base model. This foundation model is used in both data pre-processing procedures.

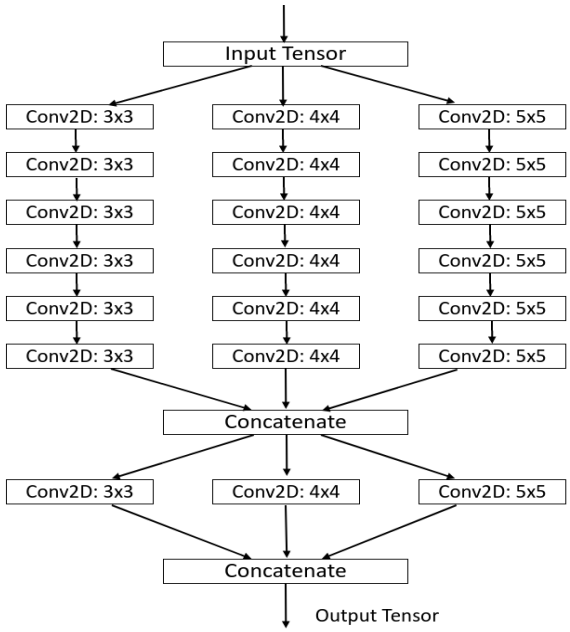

Fig. 2.

B. Neural network structure overview.

1) The fundamental model: To allow the network to learn a high number of feature levels, convolution stacked blocks are employed. A block is a five-convolution stacked design with the same kernel size. Three parallel blocks with kernel sizes of 33, 44, and 55 let the network architecture learn more different features. The outputs of three blocks are combined and routed via a similar, but shallower, architecture with only one convolution layer. Finally, to reach the appropriate output size, the output is concatenated and sent through a convolution layer. The architecture of the base model is shown in Figure 2.

$$L(C, H, S, O) = \frac{\alpha \mathrm{MSE}(C, H) + \beta \mathrm{MSE}(S, O)}{\alpha + \beta} \qquad (1)$$

2) Prepare network: The prepare network configuration is the fundamental model. It's in charge of extracting the characteristics of audio data more precisely before concatenating it with the picture. When employing data preprocessing procedure 1, the prepare network input is a 3D tensor of form 2552553. A 3D tensor of form 2552552 is used as the network input for data pre-processing procedure 2. In both cases, the desired output is a tensor of type 255255[feature maps]. The audio characteristics are extracted by the preparation network and then concatenated with the cover picture to provide the input for the hidden network.

3) Concealing network: The concealing network, like the preparing network, follows the fundamental model's architecture. The network's job is to hide the secret audio features in the cover picture so that the container image may be built, and its input is a tensor that includes both the cover image and the secret audio information. The network's weights are modified so that the output container image is identical to the cover image.

4) Reveal network: This network is responsible for decoding the container picture and recovering the original audio. We utilise a linear activation function instead of a rectified linear unit (ReLU) in the last layer of the reveal network due to the peculiarities of audio data.

The network's output is tailored to the decoding technique for each pre-processing method. A 3D tensor containing raw audio of shape 2552553, and a 3D tensor with STFT pre-process of shape 2552552, are the expected outputs. We used the Inverse Short Time Fourier Transform Algorithm (ISTFT) to recreate the original sound using the STFT method [6].

C. Error measurement

Our major aim, as previously stated, is to develop a model that can both hide and extract audio from a picture. The concealing network's goal is to hide hidden audio S in cover picture C. The newly created picture, known as the container image or hidden image H, holds information about secret audio once S has been hidden. The disclosed audio O is extracted from the original audio once the container image is delivered to the reveal network. To compare the disparities between the cover picture and the container image, as well as between the hidden audio and the revealed audio, we use the mean square error (MSE) metric. The loss function used is a weighted sum of the mean squared errors of the hiding and reveal networks.

The mean square error between the cover picture C and the hidden image H is MSE(C, H), whereas the mean square error between the secret audio S and the revealed audio O is MSE(S, O), The container picture might be of higher quality when the value of is bigger. On the other hand, as the value of grows, the quality of the recovered audio improves. During the training phase, the network's parameters are changed to minimise L (C, H, S, O).

IV. CONFIGURATION OF THE DATASET AND THE SYSTEM

A. Set of data

We put our findings to the test using two datasets: the Vivos dataset and a popular set of picture data described further down. Vivos is a publicly available dataset that contains 12,420 Vietnamese voices recorded at a sample rate of 16kHz. The audio is between 1 and 18 seconds long. Photos from Kaggle2 contests such as the Flower, Fruits, Dogs and Cats, and Stanford Dogs datasets, among others, are included in the picture dataset. The gathered picture collection contains up to 24,000 photographs. Eighty percent, ten percent, and ten percent of the dataset are used in the training, validating, and testing phases, respectively.

B. System configuration

Our test was done on a computer with a 3.4GHz Intel Core i5-7500 CPU, 16GB of RAM, and a GeForce GTX 1080 Ti GPU. The suggested architecture is implemented using the Keras3 framework.

V. RESULTS AND DISCUSSIONS

We utilised the sum of squares error (SSE) measure on 1000 images to verify the integrity of the cover picture before and after the secret audio was buried. To put it

another way, we compute MSE per pixel and per channel on 1000 pairs of pictures before combining the results. The MSE per pixel and each channel is calculated as follows: The MSE per pixel, per channel, if the image's height and width are H and W, respectively, is:

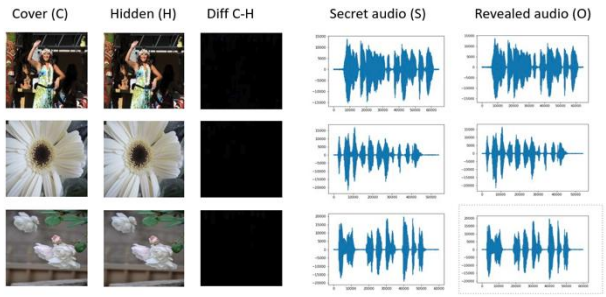

Fig. 3. STFT pre-processing based results.

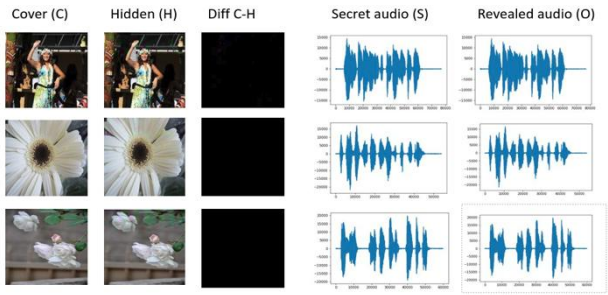

Fig. 4. raw data based results in pre-processing

TABLE I
SSE (smaller is better) and correlation coefficient (greater is better) with raw and STFT pre-processing techniques

Pre-process	α/β	SSE	Average of correlation
Method-1: Raw	15 / 1	0.3744	99.83%
	10 / 1	2.0379	99.30%
	1 / 1	2.4072	99.84%
	1 / 2	10.7196	99.74%
	1 / 10	14.8777	99.90%
Method-2: STFT	10 / 1	0.0192	99.43%
	2 / 1	0.0135	99.91%
	1 / 1	0.0334	99.85%
	1 / 2	0.0539	99.86%
	1 / 10	0.1393	99.93%

$$\mathrm{SSE} = \sum_{1}^{1000} \mathrm{MSE}_{per\_pixel,\ per\_channel} \qquad (2)$$

where y, y corresponds to the values of each pixel in each colour channel in the cover and container images, and x, x corresponds to the values of each pixel in each colour channel in the cover and container images. The Pearson correlation coefficient (PCC) is a method for determining the integrity of audio data after it has been hidden and revealed. The linear link between two data distributions is measured by the Pearson correlation coefficient. The following formula is used to compute the correlation coefficient:

$$\mathrm{MSE}_{per\_pixel,\ per\_channel} = \frac{1}{3 \times H \times W} \sum (y - \hat{y})^2, \qquad (3)$$

where mx is the vector x's mean and my is the vector y's mean.

A correlation's value runs from -1 to +1, with 0 representing no relationship. A linear connection is implied by a correlation of -1 or +1. When the correlation coefficient is 1 or 100 percent, two audio files are deemed identical [11], [16].

$$\mathrm{r} = \frac{\sum (x - m_x)(y - m_y)}{\sqrt{\sum (x - m_x)^2 \sum (y - m_y)^2}} \qquad (4)$$

The outcome of the evaluation

Tables I and II show the outcomes of the experiments. It demonstrates the application of our approaches to audio-into-image steganography. When applying STFT pre-processing, the average correlations across many parameter values are all over 99 percent, and may be as high as 99.93 percent, without sacrificing any detectable image quality. Audio correlation measurements are somewhat better with STFT pre-processing compared to the raw audio method with equal parameter values, whereas SSE metrics are significantly better. Our approaches can also be compared to the least significant bits (LSB) methodology. Some of the LSB of the cover picture is replaced with data from the hidden audio by LSB. The amount of bits that can be hidden in LSB is limited by the number of LSB bits that may change for each pixel; however, using our technique, the network can additionally conceal information in the correlations between pixels, resulting in a more realistic container picture and more hiding capacity. A 4-second audio clip may be buried within a 255255 picture using our DFT pre-processing approach. The duration of audio data can be up to 12 seconds when using raw audio data. STFT pre-processing is the obvious victor when both the quality of hiding-recovery and the duration of the source audio are taken into account.

It also includes a spectrogram XOR picture, which shows how the audio quality has declined over time.

Figures 3 and 4 show some of the outcomes of our technique, which included STFT pre-processing and raw data pre-processing.

VI. CONCLUSION

This research investigates the application of deep learning algorithms in audio-to-image steganography. Two convolutional neural networks are used in tandem to hide hidden audio in a public image and recover the true audio data from the encoded image. To make the present approach more suited for audio data, two unique audio processing methodologies are offered, as well as some neural network design adjustments. To prove the superiority of our suggested approach, extensive testing was conducted in a variety of circumstances. The results of the tests have validated the original image and audio's integrity. When compared to standard steganography methods, the duration of concealed audio is also enhanced. As a consequence, the suggested solution is suitable for solving the audio-to-image steganography problem.

REFERENCES

[1] Shumeet Baluja. Hiding images in plain sight: Deep steganography. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vish- wanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems 30, pages 2069–2079. Curran Associates, Inc., 2017.

[2] Abbas Cheddad, Joan Condell, Kevin Curran, and Paul Mc Kevitt. Digital image steganography: Survey and analysis of current methods. Signal Processing, 90(3):727 – 752, 2010.

[3] Nedeljko Cvejic and Tapio Seppanen. A wavelet domain lsb insertion algorithm for high capacity audio steganography. In Proceedings of 2002 IEEE 10th Digital Signal Processing Workshop, 2002 and the 2nd Signal Processing Education Workshop., pages 53–55. IEEE, 2002.

[4] Nedeljko Cvejic and Tapio Seppanen. Increasing robustness of lsb audio steganography using a novel embedding method. In International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004., volume 2, pages 533–537. IEEE, 2004.

[5] Kaliappan Gopalan. Audio steganography using bit modification. In 2003 International Conference on Multimedia and Expo. ICME'03. Proceedings (Cat. No. 03TH8698), volume 1, pages I–629. IEEE, 2003.

[6] Daniel W. Griffin and Jae S. Lim. Signal estimation from modified short-time fourier transform. 1984.

[7] Julio C Hernandez-Castro, Ignacio Blasco-Lopez, Juan M Estevez-Tapiador, and Arturo Ribagorda-Garnacho. Steganography in games: A general methodology and its application to the game of go. computers & security, 25(1):64–71, 2006.

[8] Chet Hosmer. Discovering hidden evidence. Journal of Digital Forensic Practice, 1(1):47–56, 2006.

[9] Nitin Kaul and Nikesh Bajaj. Audio in image steganography based on wavelet transform. International Journal of Computer Applications, 79(3), 2013.

[10] Imran Khan, Bhupendra Verma, Vijay K Chaudhari, and Ilyas Khan. Neural network based steganography algorithm for still images. In INTERACT-2010, pages 46–51. IEEE, 2010.

[11] Charles J. Kowalski. On the effects of non-normality on the distribution of the sample product-moment correlation coefficient. Journal of the Royal Statistical Society. Series C (Applied Statistics), 21(1):1–12, 1972.

[12] Charles Kurak and John McHugh. A cautionary note on image downgrading. In [1992] Proceedings Eighth Annual Computer Security Application Conference, pages 153–159. IEEE, 1992.

[13] Bin Li, Junhui He, Jiwu Huang, and Yun Qing Shi. A survey on image steganography and steganalysis. Journal of Information Hiding and Multimedia Signal Processing, 2(2):142–172, 2011.

[14] S. Hamid Nawab and Thomas F. Quatieri. Advanced topics in signal processing. chapter Short-time Fourier Transform, pages 289–337. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1987.

[15] Rully Adrian Santosa and Paul Bao. Audio-to-image wavelet transform based audio steganography. In 47th International Symposium ELMAR, 2005., pages 209–212. IEEE, 2005.

[16] STUDENT. PROBABLE ERROR OF A CORRELATION COEFFI-CIENT. Biometrika, 6(2-3):302–310, 09 1908.

[17] Dengpan Ye, Shunzhi Jiang, and Jiaqin Huang. Heard more than heard: An audio steganography method based on gan. arXiv preprint arXiv:1907.04986, 2019.

[18] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In Proceedings of the European Conference on Computer Vision (ECCV), pages 657–672, 2018.