THInImg: Cross-modal Steganography for Presenting Talking Heads in Images

Lin Zhao[1]* Hongxuan Li[1] Xuefei Ning[2] Xinru Jiang[3]
[1]TKLNDST, CS, Nankai University
[2]Department of Electronic Engineering, Tsinghua University
[3]Department of Computer Science, University of British Columbia
{lin-zhao,hxli}@mail.nankai.edu.cn; foxdoraame@gmail.com; xrjiang@student.ubc.ca

Abstract

Cross-modal Steganography is the practice of concealing secret signals in publicly available cover signals (distinct from the modality of the secret signals) unobtrusively. While previous approaches primarily concentrated on concealing a relatively small amount of information, we propose THInImg, which manages to hide lengthy audio data (and subsequently decode talking head video) inside an identity image by leveraging the properties of human face, which can be effectively utilized for covert communication, transmission and copyright protection. THInImg consists of two parts: the encoder and decoder. Inside the encoder-decoder pipeline, we introduce a novel architecture that substantially increase the capacity of hiding audio in images. Moreover, our framework can be extended to iteratively hide multiple audio clips into an identity image, offering multiple levels of control over permissions. We conduct extensive experiments to prove the effectiveness of our method, demonstrating that THInImg can present ***up to 80 seconds of high quality talking-head video (including audio) in an identity image with 160×160 resolution****.*

1. Introduction

Steganography is the art of discreetly embedding secret signals into overt signals, known as cover media, to ensure that only authorized recipients possess the capability to extract and decipher these concealed signals from the cover media [4, 22, 24, 46, 49]. In the process of steganography, cross-modal steganography means that the secret signals and the cover signals have different modalities. It is necessary for cross-modal steganography to unify the data from different modalities into a consistent format before concealing secret data, followed by the translation of the data format back to its original mode during the recovery process. However, how to align the data of different modalities and embed a large amount of secret signals in cover signals inconspicuously remains a significant challenge nowadays.

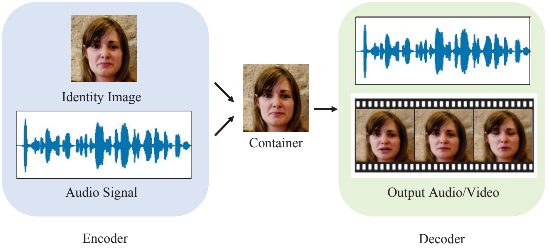

Figure 1. The pipeline of THInImg to hide lengthy audio data in identity images and generate talking head videos (with audio) from the images.

Cross-modal steganography encompasses various categories, and hiding audio in images is one of them. Several research efforts have ventured into exploring this aspect. Traditional methods typically hide data in the spatial or transformation domains through manual design. The least significant bits (LSB) algorithm, being the most commonly utilized among these methods, leverages the least significant bits of the cover media to conceal information. Aagarsana *et al.* [1] apply this algorithm to hide audio signals in RGB images in spatial domain. Similarly, Hemalatha *et al.* [15] use this algorithm to hide audio signals in YCbCr images in the transform domain using integer wavelet transform. However, as the limited number of insignificant bits in the cover media, only a small number of bits can be utilized when using the LSB algorithm. Consequently, the effective hiding capacity of secret information remains considerably limited. Recently, some deep learning methods have been proposed in cross-modal steganography to improve the hiding effect. For instance, Huu *et al.* [16] achieve embedding a 4 second audio clip with short-time fourier transform (STFT) format in a 255×255 image by using deep convolutional neural network (DCNN) model. Gandikota *et al.* [13] utilize the generative adversarial networks (GAN) model to hide a 2 second audio clip in a 128×128 image. It is clear that the length of concealed au-

*Corresponding author

dio in the aforementioned works are not enough. This is because of two primary reasons. First, audio data typically has a larger size compared to image data. For instance, the bit count of a 20-second raw audio clip is approximately 2.33 times that of a 160×160 image. Second, the human auditory system (HAS) demonstrates a heightened sensitivity to fluctuations in audio frequencies and the presence of noise. Therefore, the concealment and recovery of lengthy audio data are difficult.

In this paper, with the aim of expanding the generalization and usefulness of our system, we not only focus on hiding and recovering lengthy audio data in images, but also extend to decode human talking heads. In this way, our system can be applied in various scenarios, including but not limited to: 1) We enable covert video communication among multiple people/parties, ensuring privacy and content confidentiality. Furthermore, with solely image transmission instead of video, our system reduces video communication's bandwidth demands. 2) Visually indistinguishable facial photos can be personalized for different viewers by using different decoders, like *e.g.* "working" or "leisure" themed videos. 3) We can embed copyright information into images to provide evidence of image source and ownership. As shown in Fig. 1, our system consists of encoder and decoder for encoding and decoding the audio-visual information. By using vast prior knowledge of human face, talking head generation methods [7,8,48] can obtain vivid talking head videos by an identity image and speech. We apply a talking head generation model in the decoder to take the advantage of this characteristic. To increase the hidden audio capacity, we innovatively propose a hiding-recovering architecture, which compressing data during the steganography process, and the compressed data (acoustic features) approach to **2/7** of its original size. This compression employs non-uniform techniques, aligning with human nonlinear auditory perception. Consequently, while increasing capacity, audio quality restoration is ensured.

To support multiple access levels in our system, we further propose to hide various audio clips in the image iteratively, which enables the reconstruction of diverse talking-head videos and their corresponding audio at varying access levels. Extensive experiments are conducted on both single-speaker and multi-speaker databases to demonstrate the effectiveness of our system. The results indicate that each container image can be decoded to different lengths (up to 80 seconds) of high quality videos with audio.

In summary, our main contributions are as follows:

- To the best of our knowledge, we are the first to hide lengthy audio data (and subsequently decode talking head video) inside an image.
- We present a hiding-recovering architecture to significantly increase the capacity of hiding audio in images, enabling our THInImg to decode high-quality videos (including audio) of various lengths, up to 80 seconds for each image.
- Our THInImg system provides support for multiple access levels, enabling the iterative embedding of multiple audio clips into a single image, different talking-head videos can be decoded at each level.

2. Related Work

2.1. Cross-Modal Steganography

In recent years, numerous steganography methods have emerged, enabling the concealment of secret information in a diverse range of data types such as images, videos, and audio. Among them, images are the most commonly used ones that can be used as cover media [4,5,36]. Researchers have been able to embed many different forms of information in cover images. For example, Tancik *et al.* [34] introduce a learned steganographic algorithm to enable hiding hyperlink bit-strings in images. Watermarks are discreetly embedded in images to safeguard copyright, and they find extensive usage in interactive mobile applications [11,27]. Moreover, the video data is frequently favored as a kind of cover media for cross-modal steganography due to its substantial size and the statistical intricacy of its diverse features [26]. For example, Wengrowski *et al.* [40] develop a deep photographic steganography network to obscure light field messaging in the video. Besides, Lu *et al.* [23] hide an image set in videos to protect the biometric data during transmission for secure personal identification. Noteworthy, various audio communication solutions have been widely applied in the industry, making audio data a suitable type of cover media. Cui *et al.* [9] propose a framework to obscure images into audio, which is the first to link both image and audio media in steganography. Yang *et al.* [43] first accomplish the concealment of videos in cross-modal steganography by compressing the video data and embedding it into audio signals. In this paper, we hide talking heads in images by using human-face properties, which is first to embed lengthy audio-visual information in cover .

2.2. Audio-Driven Talking Head Generation

With the rapid development of deep learning techniques, many approaches to generating audio-driven talking head have been introduced in recent years [6,7,17,30–32,37,45].

At the early stage of the talking head study, researchers focus on generating talking-face videos by driving face regions cropped in video frames. For instance, Chung *et al.* [7] first suggest to create a video of the target face lip-synced with the audio. However, this idea does not consider the time-dependency across video frames, resulting in abrupt lip movements. After that, Song *et al.* [31] incorporate the time-dependency of image and audio features in the

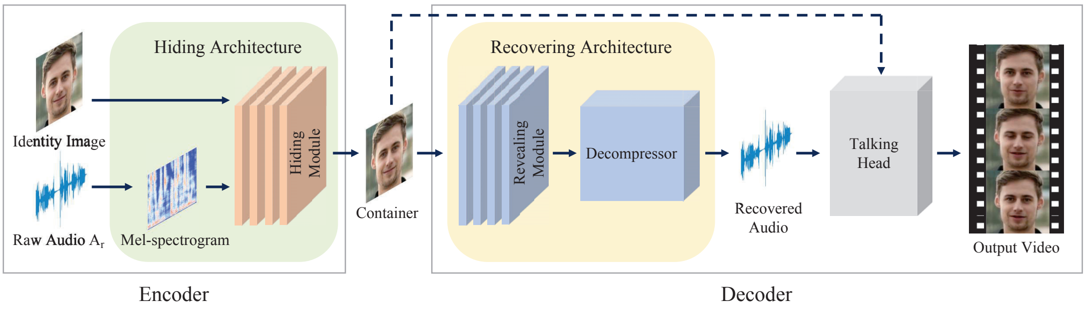

Figure 2. The overall framework of THInImg. The encoder generates container images. Talking head videos are generated in the decoder.

recursive units of their generation network. Nevertheless, the methods illustrated above consider only the face area, without the head movement and background, which makes the generated video unrealistic.

Later, to improve the realism of the generated video, the video generation process encompass not only the facial region but also the person's neck, hair, and the background. For example, Suwajanakorn *et al.* [32] focus on lip synthesis first then embed it into the original video to get a real video of Obama. Indeed, the model's usability is constrained to a particular individual and it necessitates a substantial amount of training data in the form of long videos specific to that person. To address this limitation, Zhou *et al.* [48] and Yi *et al.* [45] develop more generalized models by integrating landmarks and 3D facial data, respectively. In our work, we apply the audio-driven talking head generation model to decode videos from the identity image.

Recently, there are also some works about talking-head video compression [2, 14, 35, 39]. Cleverly leveraging generative models, Wang *et al.* [39] only needs to store a few frames and the key facial landmark information for each frame during the transmission process. Furthermore, video interpolation and super-resolution techniques are incorporated at the receiving end to further reduce the required number of bits [2]. In contrast to them, we not only reduce the video bit rate to the size of an image, but also enable covert communication.

3. Proposed Method

3.1. Encoder-Decoder

Our THInImg can be seen as a new cross-modal steganography method to encode lengthy audio and decode talking head videos solely in an identity image. As shown in Fig. 2, THInImg consists of encoder and decoder.

Encoder. In the encoder, our primary purpose is to generate the container image I_c by hiding the speech content in the identity image I_i. Formally, the encoder of our THInImg system can be expressed as:

$$I_c = E(A_r, I_i), \tag{1}$$

where $E(\cdot)$ denotes the operation of the encoder, and A_r is the raw audio.

Decoder. Decoding videos from a given container image requires two steps: recovering the audio signal and adapting a talking-head animating model $G(\cdot)$ to generate videos. The formula is as follows:

$$O_t = D(I_c) = G(Rec(I_c)), \tag{2}$$

where $D(\cdot)$ is the decoder, $Rec(\cdot)$ represents the recovering architecture that performs the recovering operation, and O_t is the generated result video. In details, we apply the architecture in [48], which is the state-of-the-art image-driven generation approach available, to produce plausible talking-head animations with facial expressions and head motions.

3.2. The Hiding-Recovering Architecture

As depicted in Fig. 2, we propose a hiding-recovering architecture to efficiently hide lengthy audio data, including compression-decompression of the raw audio and embedding-revealing of the compressed acoustic features. In the process, we need to compress the raw audio before embedding and decompress it after revealing.

Audio compression-decompression. Most existing audio-related steganography models either directly reshape the raw audio data into audio tensors [9] or use the short-time Fourier transform (STFT) to calculate audio tensors [21, 44]. We innovatively apply non-uniform compression $Com(\cdot)$ to the audio, aligned with the non-linear auditory perception of humans, to significantly reduce the amount of data that needs to be hided while ensuring minimal audio quality loss. By this way, we choose the Mel-spectrogram be the compressed acoustic features, which can better map the human auditory perception [10]. After getting the Mel-spectrogram, by splitting and stitching it into c channels (the calculation is in Sec. 4.4), we get A_s as the input to be embedded, in which the length and width are consistent with the identity image. Thus the formula of audio compression is:

$$A_s = Comp(A_r). \tag{3}$$

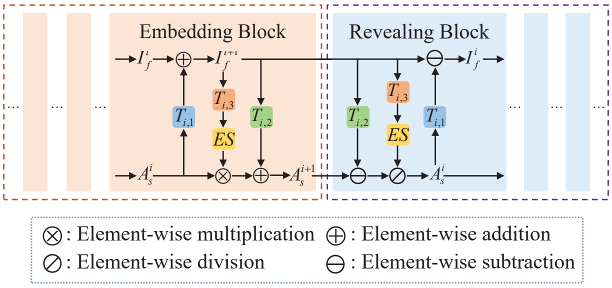
Embedding Block
Revealing Block
I_f^i
I_f^{i+1}
$T_{i,1}$
$T_{i,2}$
$T_{i,3}$
ES
A_s^i
A_s^{i+1}
⊗ : Element-wise multiplication
⊕ : Element-wise addition
⊘ : Element-wise division
⊖ : Element-wise subtraction

Figure 3. The detail composition of one of 8 embedding-revealing blocks in our uniform module.

Obviously, when obtain the revealed audio signal A_{r_R} from the container image I_c, we need to decompress the raw audio waveforms A_e from the Mel-spectrogram, and apply the DCNN model - WaveNet vocoder [33] as decompressor. Similarly, the formula of audio decompression is:

$$A_e = Decomp(A_{r_R}). \tag{4}$$

The Embedding-revealing Module. We introduce an invertible neural networks (INN) based network [3, 12, 19, 22, 28, 41, 42, 47] to embed and reveal A_s uniformly, which has been demonstrated for efficient image processing [19, 22, 47]. Our embedding and revealing module is invertible, and it conducts both forward and backward propagation operations synchronously for enabling efficient training. When the module is propagating forward, we embed A_s in I_i to output I_c as Equ. (4). Similarly, when the module is propagating backward, we reveal the Mel-spectrogram A_{s_R} and the facial image I_{i_R} from I_c:

$$(A_{s_R}, I_{i_R}) = Rev(I_c), \tag{5}$$

where $Rev(\cdot)$ represents the module that performs the revealing operation.

In more detail, the INN-based module consists of several invertible embedding-revealing blocks (8 in our method). The specific structure of each block is shown in Fig. 3. When the i-th block in our network propagates forward, it performs the operations in the embedding block:

$$\begin{aligned} {I_i}^{i+1} &= {I_i}^{i} + E_{i,1}({A_s}^{i}), \\ {A_s}^{i+1} &= {A_s}^{i} * ES(E_{i,3}({I_i}^{i+1})) + E_{i,2}({I_i}^{i+1}), \end{aligned} \tag{6}$$

where ${A_s}^i$ and ${I_i}^i$ are the input of the i-th block. $E_{i,j}(\cdot)$ represents the j-th ($j = 1, 2,$ or 3) encoding module in the i-th block, which can be any form of neural network architecture. In our experiment, $E_{i,k}$ is the residual block. $ES(\cdot)$ refers to the sigmoid function followed by the exponent. We use $ES(\cdot)$ as a multiplier to strengthen the encoding ability. Likewise, when propagating backward, the network performs the operations in the revealing block:

$$\begin{aligned} {A_s}^{i} &= \frac{{A_s}^{i+1} - E_{i,2}({I_i}^{i+1})}{ES(E_{i,3}({I_i}^{i+1}))}, \\ {I_i}^{i} &= {I_i}^{i+1} - E_{i,1}({A_s}^{i}). \end{aligned} \tag{7}$$

Here ${A_s}^{i+1}$ and ${I_i}^{i+1}$ are the input of the i-th block in the backward operations.

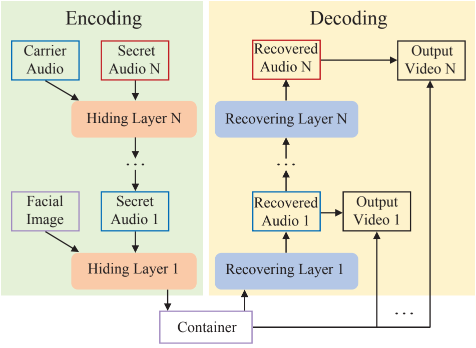

Figure 4. The diagram of nested embedding architecture with N iterations. The left part is for encoding and the right is for decoding, and the same color boxes mean same contents.

3.3. Nested Embedding Architecture

As shown in Fig. 4, we can cascade multiple embedding-revealing modules to form a nested embedding architecture, which enables the iterative hiding of various audio clips for giving different users different access levels. To ensure the effectiveness of the entire system simultaneously, we adopt an end-to-end training approach for the architecture. Here, we detail an example of a two-layer nested embedding architecture, which can be easily extended to multiple layers. The first layer network E_1 serves the same purpose as the above base module in Sec. 3.2, embedding the speech information of the first iteration A_{s_1} into the identity image I_i to get the container image I_c. Moreover, A_{s_1} acts as the container in the second layer network E_2 to embed the speech information of the second iteration A_{s_2}. As before, the network is propagating forward during encoding:

$$\begin{aligned} A_{s_1} &= E_2(A_{s_2}, A_c), \\ I_c &= E_1(A_{s_1}, I_i), \end{aligned} \tag{8}$$

where A_c represents the cover media of E_2. When I_c needs to be decoded, the network conducts backward propagation:

$$\begin{aligned} (A_{{s_1}_R}, I_{i_R}) &= E_{1_R}(I_c), \\ (A_{{s_2}_R}, A_{c_R}) &= E_{2_R}(A_{{s_1}_R}), \end{aligned} \tag{9}$$

where $A_{{s_1}_R}$ and $A_{{s_2}_R}$ denote the speech information decoded by the first and second iterations of I_c. E_{1_R} and E_{2_R} represent the recovering operation of the network during the decoding process, while I_{i_R} and A_{c_R} are the obtained cover media, respectively.

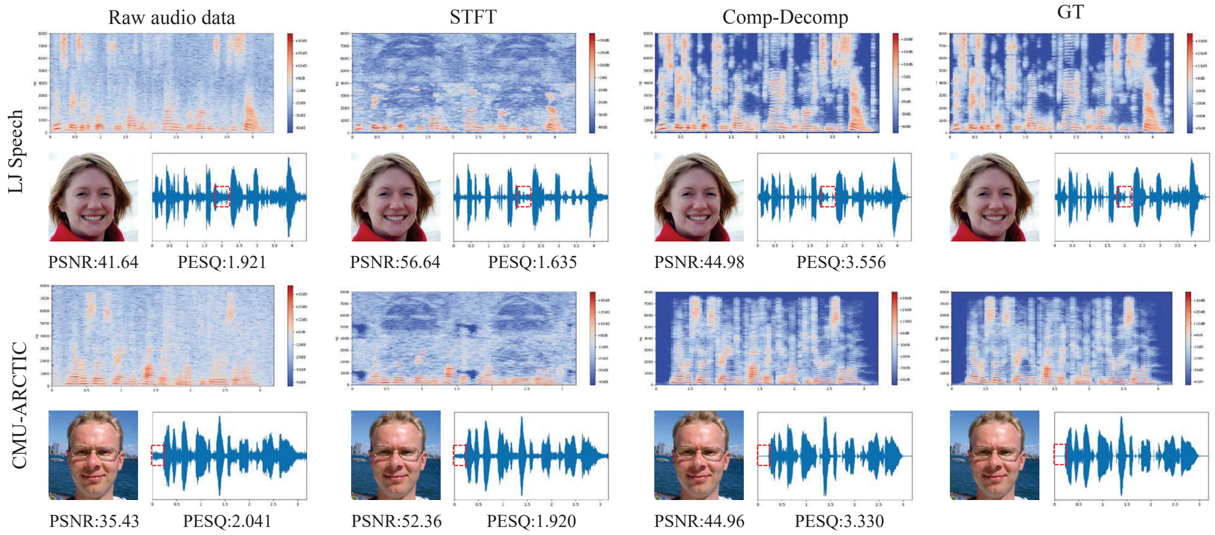

Figure 5. The visual results of the container image and recovered audio from different methods. ”GT” represents reference audio and the original identity image. From the red dashed box, only the results with audio compression-decompression avoid generating excess noise.

3.4. Loss Function

The purpose of the embedding-revealing module in Sec. 3.2 is twofold. Firstly, both I_c and I_{c_R} need to be as similar as possible to the identity image I_c, so their loss functions can be expressed separately as:

$$\ell_C = \frac{1}{N}\sum_{i=1}^{N}\|I_c - I_i\|^2, \tag{10}$$

$$\ell_I = \frac{1}{N}\sum_{i=1}^{N}\|I_{i_R} - I_i\|^2. \tag{11}$$

Here N represents the number of the training images. In addition, the A_{s_R} recovered from I_c should be consistent with the original A_s, which can be expressed by the following formula:

$$\ell_A = \frac{1}{N}\sum_{i=1}^{N}\|A_{s_R} - A_s\|^2. \tag{12}$$

Therefore, the final training loss is defined as a weighted sum of the losses mentioned above:

$$\ell_{tr} = \lambda_C\ell_C + \lambda_I\ell_I + \lambda_A\ell_A, \tag{13}$$

where λ_C, λ_I and λ_A refer to the balanced weights. To obtain the container image I_c and recovered Mel-spectrogram A_{s_R}, we set $\lambda_C = \lambda_A = 32$, $\lambda_I = 1$ during training.

The architecture in Sec. 3.3 contains two embedding-revealing modules, both of which employ the loss function described above and perform end-to-end training. Therefore, the loss of the architecture during training can be further expressed as:

$$\ell_{tr} = \ell_{tr_1} + \ell_{tr_2}, \tag{14}$$

where ℓ_{tr_1} and ℓ_{tr_2} represent the loss of the two embedding-revealing modules, respectively.

4. Experiments

4.1. Datasets

We use a facial image database, a single-person speech database, and a multi-person speech database in our experiments.

FFHQ The FFHQ Database [18] consists of 70,000 high-quality identity images and contains considerable variation in terms of age, ethnicity, and image background.

LJ Speech The LJ Speech Database consists of 13,100 short audio clips of passages from 7 non-fiction books read by a single speaker. The audio clips vary in length from 1 to 10 seconds and have a total utterance duration of approximately 24 hours.

CMU-ARCTIC The CMU-ARCTIC Database [20] is a multi-person speech database. We use speech data of 7 speakers in the database: bdl, slt, jmk, awb, rms, clb, and ksp. The total number of utterances is about 1,132 per speaker, and the total length is about 1 hour per speaker.

4.2. Metrics

We employ Perceptual Evaluation of Speech Quality (PESQ) [29] as a quantitative metric to objectively assess audio quality, which ranges from -0.5 to 4.5. The Perception based Image Quality Evaluator (PIQE) [38] is used to assess the generated talking-head video quality. The range of PIQE metric is 0 to 100, with lower values indicating higher video quality. We incorporate Mean Opinion Score (MOS) to obtain human evaluations regarding audio-visual

Database	LJ Speech				CMU-ARCTIC			
Form\0∼10s	Container image		Extracted video		Container image		Extracted video	
	PSNR	SSIM	PESQ	PIQE	PSNR	SSIM	PESQ	PIQE
Raw audio	41.50	0.979	1.855	46.87	38.08	0.965	2.026	47.53
STFT	**55.94**	**0.999**	1.697	46.77	**53.20**	**0.998**	2.132	**46.23**
Comp-Decomp	41.50	0.979	**3.639**	**46.59**	44.66	0.989	**3.215**	**46.23**

Table 1. Quantitative comparison with different methods, audio lengths ranging from 0∼10s for each image.

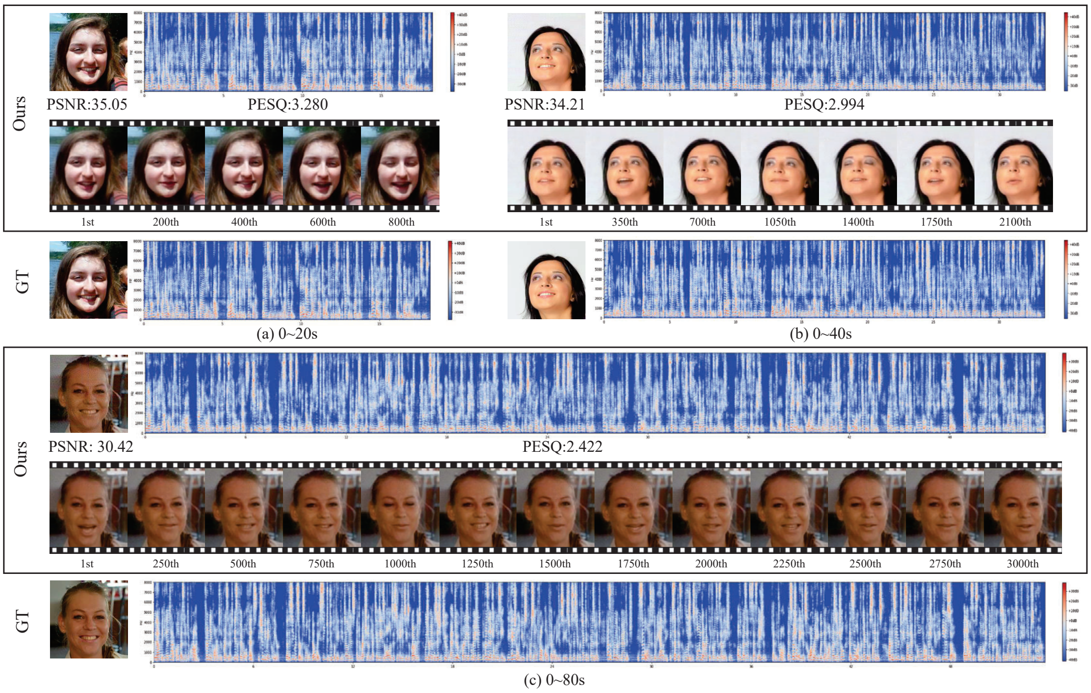

Figure 6. The decoded audio and video frames corresponding to different audio lengths in the LJ Speech database.

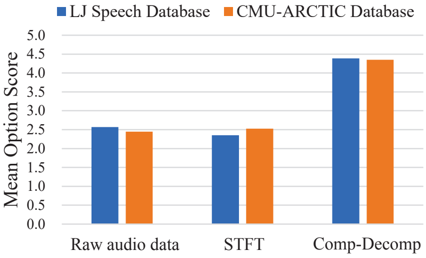

Figure 7. MOS values comparison across methods, with separate statistics for generated talking-head videos from two databases.

information quality. Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Metric (SSIM) metrics are utilized to evaluate the visual quality of container images.

4.3. Implementation Details

We conducted two sets of experiments: One is using FFHQ images and audio clips from the single-person database LJ Speech, the training set consisted of 12,522 audio clips, and the testing set comprised 578 audio clips. Each paired with an equivalent number of images. Similarly, the other is using equal number of images from FFHQ database and audio clips from multi-person speech database CMU-ARCTIC, a training set of 7,580 utterances and a testing set of 350 utterances are utilized. To ensure fairness, an equal number of utterances were randomly selected from each person in the CMU-ARCTIC database for training.

In addition, to verify effectiveness of the nested embedding architecture, we conduct experiments on LJ Speech database using a two-layer architecture as an example. The first half of the training and testing sets are for the first layer network, and the second half is for the second layer.

Input The images are resized to 160 × 160 resolution and

Database	LJ Speech				CMU-ARCTIC			
Time range	Container image		Extracted video		Container image		Extracted video	
	PSNR	SSIM	PESQ	PIQE	PSNR	SSIM	PESQ	PIQE
0∼20s	39.72	0.970	3.370	46.68	40.44	0.976	3.107	46.43
0∼40s	30.85	0.895	3.111	48.14	37.00	0.954	2.880	47.18
0∼80s	28.25	0.811	2.482	45.92	30.03	0.893	2.382	49.23

Table 2. Quantitative results of embedding different ranges of audio lengths in the THInImg system.

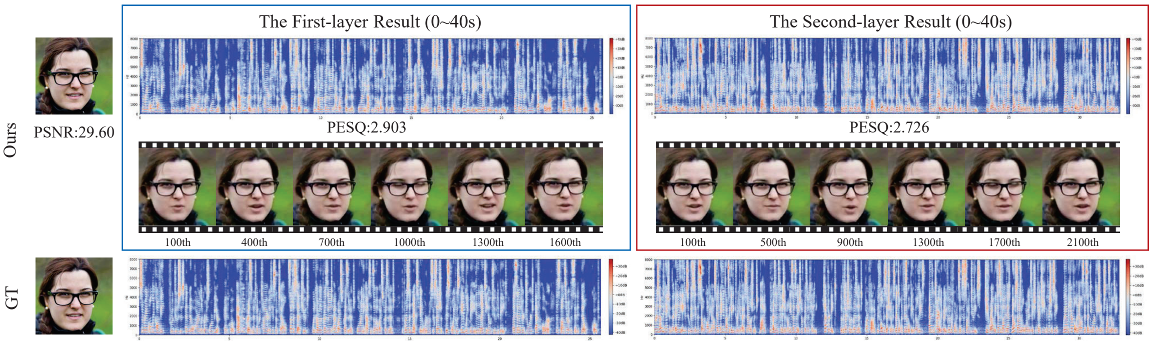

Figure 8. Visual results of the double-layer nested embedding architecture in hiding 0∼80s audio. Results for other audio length ranges are in the supplementary material.

the audio clips are adjusted to 16kHz. For the processing of the audio, spectrum is represented by applying the STFT with 1,024 FFT frequency bins and a sliding window with a shift 256. The number of Mel filters for compressing is 80.

Training The proposed algorithm is implemented in Pytorch [25], and an Nvidia 2080Ti GPU is used for acceleration. We train the decompressor for 2000 epochs in all experiments, while the embedding-revealing module and its double-layer nested hiding architecture for 100 epochs. The ADAM optimizer is applied for all networks, while the learning rate is set to 2e-4 and 1e-2.

4.4. Performance Experiments

Verification of audio compression-decompression effectiveness. We demonstrate the effectiveness by comparing the hiding-recovering architecture with two methods without compression. The architectures of two methods are applied with the raw audio data and STFT, respectively. Between them, we regard the architecture utilizing raw audio data as the baseline. The architecture using STFT, which is applied in [13, 16], as a method to improve the baseline.

To enable each image to obscure different lengths of speech, we randomly select the audio length corresponding to each image from 0 to 10s both during the training and testing processes. We reshape each of the three formats of data into a tensor with the size of $h \times w \times c$, where $h = w = 160$, whose value is the same as the image size. The STFT spectrum is complex with a form of $a + bi$, so we need to embed both the real part a and the imaginary part b. Adjusting different window sizes of the STFT spectrum can control the size of the input tensor as $h \times w \times c$. After the standardization, we get $c = 2$ for the Mel-spectrogram, $c = 7$ for the raw audio and $c = 4$ for the STFT spectrum.

Analysis of quantitative results. The results on both single-speaker and multi-speaker databases are presented in Tab. 1. We can see that only the results obtained using Mel-spectrogram as input achieve high scores for the recovered audio. The container image and the generated talking-head videos obtained using these three audio formats can each achieve a satisfactory score. The PSNR and SSIM values of the STFT format are higher than the other two methods, but the values of its PESQ are unsatisfactory.

Analysis of qualitative results. We present the visual results of container images and recovered audio in Fig. 5. The visualization of the recovered audio includes its waveform plot in the time domain and its Mel-spectrogram plot in the frequency domain. As can be seen, it is challenging to discover the difference between the container image of each method and the original facial image with the naked eye. However, for the recovered audio, the waveform plots of the two methods – using raw format or STFT format – are different from that of the reference audio. In addition, all waveforms are noisy except for the results obtained using the Mel-spectrogram as input, especially as shown by the dashed boxes Fig. 5, which represents gaps in speech. Mel-spectrogram plots visually indicate that only the third column method produces similar output with reference audio.

Naturalness MOS. To obtain the subjective mean opinion

Database	LJ Speech					
Time range	Container image		First video		Second video	
	PSNR	SSIM	PESQ	PIQE	PESQ	PIQE
0∼20s	39.60	0.967	3.521	46.44	3.414	46.58
0∼40s	36.32	0.938	3.424	47.42	3.280	47.56
0∼80s	30.87	0.864	2.912	49.58	2.857	49.63

Table 3. Quantitative results of hiding different ranges of audio lengths in the THInImg. “First video” and ”Second video” mean the extracted talking head videos of the first layer and the second layer.

score (MOS) of each method, we randomly selected 2 audio clips per person in the CMU-ARCTIC database and 6 audio clips in the LJ Speech database to generate the videos, making a total of 20 talking-head videos for rating. We invited 30 participants who rated the samples on a scale of 0∼5 with 0.5 point increments. The testers rate the talking-head videos by a combination of audio quality and video reality. As shown in Fig. 7, the highest values are reached when using the Mel-spectrogram format in both databases.

4.5. Nested Embedding and Capacity Study

We test **the THInImg capacity for both the base and the double-layer nested embedding architectures**. We splice the audio clips in two databases separately to perform experiments on embedding speech information of different ranges. The quantitative results of the base single-layer model are shown in Tab. 2. As the audio lengths increase, the number of channels c in the network increases, performance decreases for all metrics. It can be seen more visually in Fig. 6 for single speaker and Fig. 8 for multiple people that when audio lengths range is less than 40s, the results are satisfactory for both container images and recovered audio. When the length reaches 80 seconds, although the audio is noiseless and the Mel-spectrogram plot is approximately the same, few artifacts in the container image.

The results are shown in Tab. 3 and Fig. 8 for the two-layer nested embedding architecture. The audio ranges in the first and second layers are the same, and the audio quality in the deep layer is slightly worse than that in the shallow layer. As mentioned above, the effect is satisfactory but worsens as the audio lengths increase.

We show that our THInImg can hide and reveal 80 seconds speech with guaranteed quality through experiments. The recovered audio and the final generated videos of talking heads are displayed in the supplementary material.

4.6. Discussion about number of nested layers

We observe two interesting phenomena. First, through Tab. 2 and Tab. 3, when the maximum audio lengths of hiding are 40s and 80s, the qualities of the image and audio in the two-layer nested architecture are better than the base single-layer model. It is because when hiding same length audio, the two-layer nested architecture reduces concealed audio length in each layer, compared to the base single-layer model. Therefore, when concealing audio over a longer range, the two-layer nesting performs more effectively. It can be inferred that models with higher nested layers tend to exhibit better performance when the nesting depths are in an appropriate range. Second, as mentioned in the results of the nested architecture, the quality of deep layer is worse than shallow layer. It is easily inferable that the quality will be degraded when the number of layers reaches a certain threshold due to the loss incurred from training a multi-layer architecture exceeds the benefits derived from less data embedded each layer mentioned in the first point. As the number of layers continues to increase, excessively high model loss will result in notably poor quality.

Considering the above two points collectively, the selection of the number of layers for network nesting is a topic worthy of discussion when hiding audio of a specific length. In the future, we will explore several sub-questions arising from the aforementioned issue, including: 1. Selection of weights for different layers in multi-layer nesting. 2. The number of layers can be nested until quality severely degrades. 3. Regarding our system, what is the optimal number of layers for achieving the best results, and does this choice allow more data (longer than 80s) to be hidden?

5. Conclusion

This paper proposes THInImg, a cross-modal steganograply method to hide lengthy audio data and decode the talking-head videos with audio (up to 80 seconds) in 160x160 identity images. There is an encoder and a decoder in THInImg for encoding audio and decoding the videos. In the hiding-recovering pipeline, a novel architecture was introduced to simultaneously increase the length of concealed audio while ensuring audio quality. Furthermore, We extended the structure to allow for nested embedding, providing different access priorities for users. Numerous experiments have been conducted to demonstrate the effectiveness of our method.

References

[1] BG Aagarsana, T Kirthika Anjali, and Mr S Sivakumar Kirthika. Image steganography using secured force algorithm for hiding audio signal into colour image. *IRJET access*, 5, 2018. 1

[2] Madhav Agarwal, Anchit Gupta, Rudrabha Mukhopadhyay, Vinay P Namboodiri, and CV Jawahar. Compressing video calls using synthetic talking heads. *arXiv preprint arXiv:2210.03692*, 2022. 3

[3] Lynton Ardizzone, Carsten Lüth, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Guided image generation with conditional invertible neural networks. *arXiv preprint arXiv:1907.02392*, 2019. 4

[4] Shumeet Baluja. Hiding images within images. 42(7):1685–1697, 2019. 1, 2

[5] Yambem Jina Chanu, Kh Manglem Singh, and Themrichon Tuithung. Image steganography and steganalysis: A survey. *IJCV*, 52(2), 2012. 2

[6] Lele Chen, Ross K Maddox, Zhiyao Duan, and Chenliang Xu. Hierarchical cross-modal talking face generation with dynamic pixel-wise loss. In *CVPR*, pages 7832–7841, 2019. 2

[7] Joon Son Chung, Amir Jamaludin, and Andrew Zisserman. You said that? *arXiv preprint arXiv:1705.02966*, 2017. 2

[8] Daniel Cudeiro, Timo Bolkart, Cassidy Laidlaw, Anurag Ranjan, and Michael J Black. Capture, learning, and synthesis of 3d speaking styles. In *CVPR*, pages 10101–10111, 2019. 2

[9] Wenxue Cui, Shaohui Liu, Feng Jiang, Yongliang Liu, and Debin Zhao. Multi-stage residual hiding for image-into-audio steganography. In *ICASSP*, pages 2832–2836. IEEE, 2020. 2, 3

[10] Steven Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. 28(4):357–366, 1980. 3

[11] Lorenzo Antonio Delgado-Guillen, Jose Juan Garcia-Hernandez, and Cesar Torres-Huitzil. Digital watermarking of color images utilizing mobile platforms. pages 1363–1366. IEEE, 2013. 2

[12] Michael Dorkenwald, Timo Milbich, Andreas Blattmann, Robin Rombach, Konstantinos G Derpanis, and Bjorn Ommer. Stochastic image-to-video synthesis using cinns. In *CVPR*, pages 3742–3753, 2021. 4

[13] Rohit Gandikota and Deepak Mishra. Hiding audio in images: A deep learning approach. In *International Conference on Pattern Recognition and Machine Intelligence*, pages 389–399. Springer, 2019. 1, 7

[14] Sindhu B Hegde, Rudrabha Mukhopadhyay, Vinay P Namboodiri, and CV Jawahar. Extreme-scale talking-face video upsampling with audio-visual priors. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 6511–6520, 2022. 3

[15] S Hemalatha, U Dinesh Acharya, and A Renuka. Wavelet transform based steganography technique to hide audio signals in image. *Procedia Computer Science*, 47:272–281, 2015. 1

[16] Quang Pham Huu, Thoi Hoang Dinh, Ngoc N Tran, Toan Pham Van, and Thanh Ta Minh. Deep neural networks based invisible steganography for audio-into-image algorithm. In *2019 IEEE 8th Global Conference on Consumer Electronics (GCCE)*, pages 423–427. IEEE, 2019. 1, 7

[17] Xinya Ji, Hang Zhou, Kaisiyuan Wang, Wayne Wu, Chen Change Loy, Xun Cao, and Feng Xu. Audio-driven emotional video portraits. In *CVPR*, pages 14080–14089, 2021. 2

[18] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019. 5

[19] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *NeurIPS*, pages 10215–10224, 2018. 4

[20] John Kominek and Alan W Black. The CMU arctic speech databases. In *Fifth ISCA workshop on speech synthesis*, 2004. 5

[21] Felix Kreuk, Yossi Adi, Bhiksha Raj, Rita Singh, and Joseph Keshet. Hide and speak: Towards deep neural networks for speech steganography. *arXiv preprint arXiv:1902.03083*, 2019. 3

[22] Shao-Ping Lu, Rong Wang, Tao Zhong, and Paul L Rosin. Large-capacity image steganography based on invertible neural networks. In *CVPR*, pages 10816–10825, 2021. 1, 4

[23] Yingqi Lu, Cheng Lu, and Miao Qi. An effective video steganography method for biometric identification. In *Advances in Computer Science and Information Technology: AST/UCMA/ISA/ACN 2010 Conferences, Miyazaki, Japan, June 23-25, 2010. Joint Proceedings*, pages 469–479. Springer, 2010. 2

[24] Chong Mou, Youmin Xu, Jiechong Song, Chen Zhao, Bernard Ghanem, and Jian Zhang. Large-capacity and flexible video steganography via invertible neural network. In *CVPR*, pages 22606–22615, 2023. 1

[25] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 7

[26] Rahul Paul, Anuja Kumar Acharya, Virendra Kumar Yadav, and Saumya Batham. Hiding large amount of data using a new approach of video steganography. In *Confluence 2013: The Next Generation Information Technology Summit*, pages 337–343. IET, 2013. 2

[27] Anu Pramila, Anja Keskinarkaus, and Tapio Seppänen. Toward an interactive poster using digital watermarking and a mobile phone camera. *Signal, Image and Video Processing*, 6(2):211–222, 2012. 2

[28] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP*, pages 3617–3621. IEEE, 2019. 4

[29] Antony W Rix, John G Beerends, Michael P Hollier, and Andries P Hekstra. Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. In *ICASSP*, volume 2, pages 749–752. IEEE, 2001. 5

[30] Linsen Song, Wayne Wu, Chen Qian, Ran He, and Chen Change Loy. Everybody's talkin': Let me talk as you want. *arXiv preprint arXiv:2001.05201*, 2020. 2

[31] Yang Song, Jingwen Zhu, Dawei Li, Xiaolong Wang, and Hairong Qi. Talking face generation by conditional recurrent adversarial network. *arXiv preprint arXiv:1804.04786*, 2018. 2

[32] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing Obama: learning lip sync from audio. *ACM TOG*, 36(4):1–13, 2017. 2, 3

[33] Akira Tamamori, Tomoki Hayashi, Kazuhiro Kobayashi, Kazuya Takeda, and Tomoki Toda. Speaker-dependent WaveNet vocoder. In *Interspeech*, pages 1118–1122, 2017. 4

[34] Matthew Tancik, Ben Mildenhall, and Ren Ng. StegaStamp: Invisible hyperlinks in physical photographs. In *CVPR*, pages 2117–2126, 2020. 2

[35] Pulkit Tandon, Shubham Chandak, Pat Pataranutaporn, Yimeng Liu, Anesu M Mapuranga, Pattie Maes, Tsachy Weissman, and Misha Sra. Txt2vid: Ultra-low bitrate compression of talking-head videos via text. *IEEE Journal on Selected Areas in Communications*, 41(1):107–118, 2022. 3

[36] Weixuan Tang, Bin Li, Shunquan Tan, Mauro Barni, and Jiwu Huang. Cnn-based adversarial embedding for image steganography. 14(8):2074–2087, 2019. 2

[37] Justus Thies, Mohamed Elgharib, Ayush Tewari, Christian Theobalt, and Matthias Nießner. Neural voice puppetry: Audio-driven facial reenactment. In *ECCV*, pages 716–731. Springer, 2020. 2

[38] N Venkatanath, D Praneeth, Maruthi Chandrasekhar Bh, Sumohana S Channappayya, and Swarup S Medasani. Blind image quality evaluation using perception based features. pages 1–6. IEEE, 2015. 5

[39] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. One-shot free-view neural talking-head synthesis for video conferencing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10039–10049, 2021. 3

[40] Eric Wengrowski and Kristin Dana. Light field messaging with deep photographic steganography. In *CVPR*, pages 1515–1524, 2019. 2

[41] Mingqing Xiao, Shuxin Zheng, Chang Liu, Yaolong Wang, Di He, Guolin Ke, Jiang Bian, Zhouchen Lin, and Tie-Yan Liu. Invertible image rescaling. *arXiv preprint arXiv:2005.05650*, 2020. 4

[42] Yazhou Xing, Zian Qian, and Qifeng Chen. Invertible image signal processing. In *CVPR*, pages 6287–6296, 2021. 4

[43] Hyukryul Yang, Hao Ouyang, Vladlen Koltun, and Qifeng Chen. Hiding video in audio via reversible generative models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1100–1109, 2019. 2

[44] Dengpan Ye, Shunzhi Jiang, and Jiaqin Huang. Heard more than heard: An audio steganography method based on GAN. *arXiv preprint arXiv:1907.04986*, 2019. 3

[45] Ran Yi, Zipeng Ye, Juyong Zhang, Hujun Bao, and Yong-Jin Liu. Audio-driven talking face video generation with learning-based personalized head pose. *arXiv preprint arXiv:2002.10137*, 2020. 2, 3

[46] Kevin Alex Zhang, Alfredo Cuesta-Infante, Lei Xu, and Kalyan Veeramachaneni. SteganoGAN: High capacity image steganography with GANs. *arXiv preprint arXiv:1901.03892*, 2019. 1

[47] Lin Zhao, Shao-Ping Lu, Tao Chen, Zhenglu Yang, and Ariel Shamir. Deep symmetric network for underexposed image enhancement with recurrent attentional learning. In *ICCV*, pages 12075–12084, 2021. 4

[48] Yang Zhou, Xintong Han, Eli Shechtman, Jose Echevarria, Evangelos Kalogerakis, and Dingzeyu Li. MakeItTalk: speaker-aware talking-head animation. *ACM TOG*, 39(6):1–15, 2020. 2, 3

[49] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *ECCV*, pages 657–672, 2018. 1