Hiding Audio in Images: A Deep Learning Approach

Rohit Gandikota(✉) and Deepak Mishra

Indian Institute of Space Science and Technology, Trivandrum 695547, Kerala, India
grohit0@gmail.com, deepak.mishra@iist.ac.in

Abstract. In this work, we propose an end-to-end trainable model of Generative Adversarial Networks (GAN) which is engineered to hide audio data in images. Due to the non-stationary property of audio signals and lack of powerful tools, audio hiding in images was not explored well. We devised a deep generative model that consists of an auto-encoder as generator along with one discriminator that are trained to embed the message while, an exclusive extractor network with an audio discriminator is trained fundamentally to extract the hidden message from the encoded host signal. The encoded image is subjected to few common attacks and it is established that the message signal can not be hindered making the proposed method robust towards blurring, rotation, noise, and cropping. The one remarkable feature of our method is that it can be trained to recover against various attacks and hence can also be used for watermarking.

Keywords: Steganography · Generative Adversarial Network · Auto-encoder · Deep learning · Watermarking · Latent representations

1 Introduction

Audio is an essential mean of communication in the modern era. Through audio, one can communicate about things that can't be expressed in a visually representative way. It is quick and easy to initiate compared to typing a text message. One can argue that audio transmission is lighter compared to huge video data that use up the bandwidth. There could arise a need for secret communication without any adversary able to eavesdrop or a need to inject an audio message in every original content that we create. Our work attempts to study and understand the requirements for such scenarios and come up with a solution using the most popular architectures currently in deep learning that has the ability to generate data, Generative Adversarial Networks (GAN) [8].

Two common notions exist for hiding information in images. In steganography, the goal is secret communication between a sender and a receiver, while an adversary person cannot tell if the image contains any message. In digital watermarking, the goal is to encode information robustly. Even though an adversary

B. Deka et al. (Eds.): PReMI 2019, LNCS 11942, pp. 389–399, 2019.
https://doi.org/10.1007/978-3-030-34872-4_43

distorts the image, the receiver can extract the message. Watermarking is typically used to identify ownership of the copyright of data. Our method can be used in either case since the message is visually indistinguishable and the network is robust towards most of the attacks. Hiding data can be achieved through techniques that use domain transforms to embed message [4,5,7,9], and others using spatial domain [14]. Recently data hiding using deep neural networks were introduced [10,11,13]. Zhu et al. [10] has introduced the adversarial component in data hiding using an encoder-decoder model to embed and extract respectively.

Most of the works [10–13] deal with hiding images which are stationary signals and 3 dimensional. In this paper, we first discuss the possibilities and limitations on embedding a raw audio signal which is a one dimensional non-stationary signal inside an image using the existing deep learning methods. Further, we demonstrate that hiding the spectrogram of the raw audio and use its frequency domain information of audio as one of the feasible options in our work rather than directly dealing with the non-stationary audio signal. Therefore, we compute the spectrogram of the audio and use the frequency domain information in data hiding instead of dealing with the non-stationary time domain information. The idea of playing in frequency domain is inspired from work by Yang *et al.* [16], who used the adversarial networks for style transfer in audio by modifying spectrogram of the audio signal.

The main contributions of this work are

- Introducing the first generic end-to-end trainable GAN, **VoI-GAN**, for steganography and watermarking that can hide audio in images.
- Introducing latent representations in GAN training for data hiding which made the learning process more efficient, simpler and faster.
- A clean extraction of hidden audio from encoded images (even when distorted), preserving all the key features of the audio like frequency, pitch, speaking rate, and subjective features.

2 Method

2.1 Problem Formulation

Data hiding is to embed a message (MSG) into a cover image (CI) and then later to be able to extract the message from the encoded image (EI). So there is a need for two processes to accomplish a successful secret communication. In this paper, we propose a novel method for the two above mentioned tasks. We propose an autoencoder to reduce the messages to latent representations, so as to help the training of the embedder which is a multi-objective GAN. This base learning of simple latent representation makes the GAN training efficient. An exclusive extractor network with adversarial training to fundamentally extract the embedded messages is also required.

VoI-GAN consists of five nets; an embedder **Emb**, discriminator for embedder **D1**, an extractor **Ex**, a secondary discriminator for extractor **D2** and another autoencoder **AE**, for actual message to latent domain transformation.

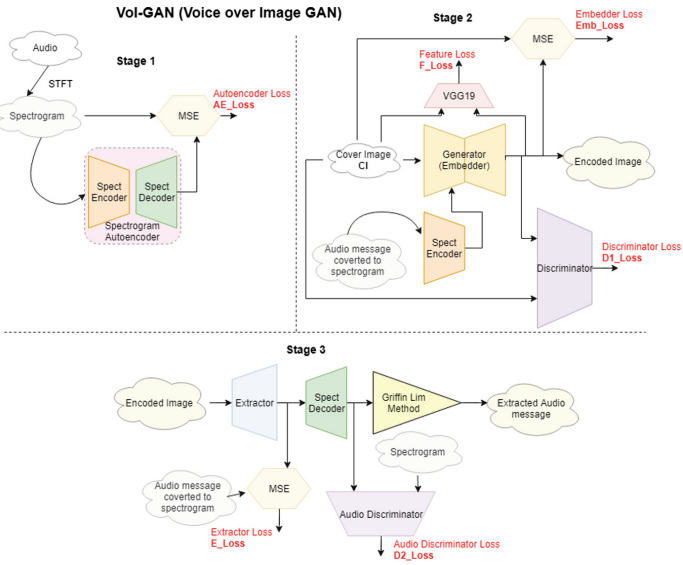

Fig. 1. Three stage explicit training stages of VoI-GAN, a network that can hide audio in images. It consists of a generator, a discriminator, an extractor, an autoencoder for spectrogram encoding and a pretrained VGG19 model for feature extraction.

It is important to note that, GANs are unstable during training [1] and may lead to distortions in output images. We propose a custom multi-objective framework with base training for VoI-GAN, which has resolved the issue of mode-collapse in GAN training.

2.2 Spectrogram Autoencoder

We believe it is since the GAN being a density estimation method, enforces the latent representation to follow simple distributions such as isotropic Gaussian as explored by Subakan *et al.* [15]. Since the autoencoder's latent space is much simpler than the base distribution, it enabled GAN in more accurate modeling over the complicated base distribution. We train an autoencoder to encode the spectrogram into the latent space. After the adversarial embedding is done, the embedded image is passed through an optional noisy channel and the message is recovered using a separately trained extractor. The output being in the latent space is converted back to the spectrogram using the decoder of **AE**. The whole process of embedding and extracting is shown in Fig. 1.

2.3 Embedder

A detailed representation of the embedder is depicted in Fig. 2. It is a convolution auto-encoder network with leaks from the encoder to decoder (U-Net structure). The message spectrogram of size $cW \times cH \times cC$, is first passed through the

encoder of the **AE** which is referred as 'message processing network' to get the latent representations of size $lW \times lH \times lC$. The embedding of this latent representation **MSG** is done through concatenation by the encoder of the **Emb** which is referred to as 'hiding network'.

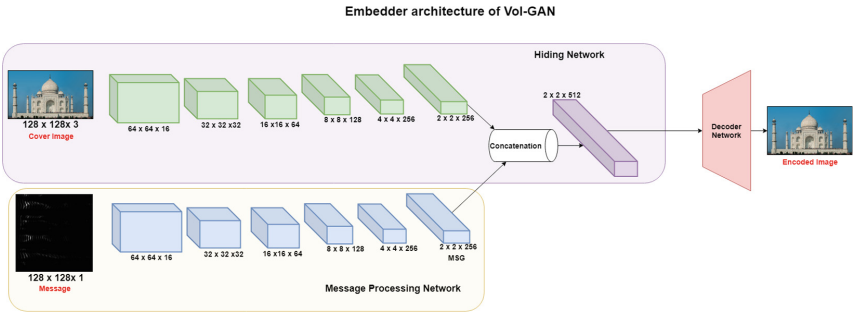

Fig. 2. Model of the embedder used in VoI-GAN. It is an auto-encoder with leaks from the encoder to decoder between the layers with similar dimensions. The encoder has two networks, message processing network and hiding network. Message processing network processes the message to concatenate with the cover image by the hiding network. It is also important to note that the message processing network is the encoder of the trained spectrogram autoencoder **AE**.

From the view of an embedder, the target is to embed the message in a visually indistinguishable way. Therefore, we design a discriminator that consists of stacks of convolutional layers with batch normalization and activation function, to evaluate whether each output of G is true or fake through lower level and semantic features. Through this discriminator network, we introduce the adversarial training into our method. To illustrate clearly, as shown in Fig. 1, the discriminator takes an image of size $cW \times cH \times cC$ as an input and outputs a single number to discriminate between the cover image and encoded image.

2.4 Extractor

To extract the hidden audio message from the encoded image, we propose an exclusive trainable CNN with U-net structure. The extractor **E**, takes EI with size $cW \times cH \times cC$ as input and is trained to extract the MSG of size $mW \times mH \times mC$ as output (as done by [3,10,12]). To get the spectrogram, **MSG** is passed through the decoder of **AE**. Optionally to improve the extraction quality, an adversarial loss can be added in addition to the pixel loss.

To preserve the features of audio like pitch, audio rate, etc., we design a secondary discriminator that distinguishes between the extracted spectrogram and the originally hidden spectrogram. For this end, we design a CNN that consists of stacks of convolutional layers with batch normalization and activation function, to evaluate the extracted spectrogram through lower level and semantic features.

Through this discriminator network, we introduce the adversarial training into the extractor. This discriminator preserves the features of the spectrogram which were ignored by the extractor when pixel loss alone was used.

2.5 Training of VoI-GAN to Encode Audio in Images

The training of VoI-GAN can be disintegrated into an implicit three-stage training process as shown in Algorithm 1. The first stage is to train the autoencoder **AE** to output an image that is similar to its input. As shown in Fig. 1, the autoencoder with parameters ϵ receives a spectrogram SPC^{in} as input and outputs a similar image SPC^{out} as output using Eq. 1.

$$AE_{Loss} = \frac{1}{cH \times cW} \sum_{i=1}^{cH} \sum_{j=1}^{cW} (SPC_{i,j}^{in} - SPC_{i,j}^{out})^2 \tag{1}$$

The second stage of training is to train the embedder to output an image that is similar to the cover image. As shown in Fig. 1, the generator with parameters θ receives a cover image CI and message image MSG as inputs and outputs an image EI as output. Using the adversarial loss, $D1_{Loss}$ in Eq. 2, alone to generate the encoded image was not sufficient as there was dilution effect near the boundary of the image. To overcome this dilution, we introduced a pixel loss between the encoded image and the cover image, referred as Emb_{loss} in Eq. 3.

$$D1_{Loss} = log\mathbf{D1}(CI) + log(1 - \mathbf{D1}(EI)) \tag{2}$$

$$Emb_{Loss} = \frac{1}{cH \times cW} \sum_{i=1}^{cH} \sum_{j=1}^{cW} (EI_{i,j} - CI_{i,j})^2 \tag{3}$$

$$Where\ EI = Embedder_{\theta}(CI)\ and\ \mathbf{D1}\ is\ Discriminator$$

Although this has improved the dilution effect, the images are still not sharp enough and had some color distortion. To achieve sharper images, we have introduced a feature loss. For this end, a pre-trained VGG19 network χ whose last pooling layer's output is considered as the features. We compared the features of encoded image EI with features of actual cover image CI and represented it as loss function F_{Loss} in Eq. 4. The feature loss has helped in restoring finer details in the encoded images. This also helped in maintaining similar texture and color information as that of the cover image making the watermark visually indistinguishable.

$$F_{Loss} = \frac{1}{fH \times fW} \sum_{i=1}^{fH} \sum_{j=1}^{fW} (\chi(EI)_{i,j} - \chi(CI)_{i,j})^2 \tag{4}$$

The third stage of training is to train the extractor, with parameters λ, to output the hidden message MSG given the current loop's EI as input. The pixel loss between the original message and the output of the encoder, E_{Loss} in Eq. 5,

along with the adversarial loss, $D2_{Loss}$, for extractor output as are designed as loss functions for extractor.

$$E_{Loss} = \frac{1}{wH \times wW} \sum_{i=1}^{wH} \sum_{j=1}^{wW} (MSG_{i,j}^{extracted} - MSG_{i,j}^{actual})^2 \tag{5}$$

The weights, $\beta_1, \beta_2, \beta_3, \beta_4, \beta_5$ were chosen in such a way that all the losses are scaled to have similar weights in the training. We have given a slightly higher weight for the F_{Loss} since it was experimentally proven to generate more appealing images. Once the hidden spectrogram is extracted, we use the Griffin-Lim's method [6] to recover the audio signal.

Algorithm 1. Three stage implicit training procedure for VoI-GAN

- Train the autoencoder with parameters ϵ such that:

$$\min_{\epsilon} \; AE_{Loss}$$

- Train the embedder with parameters θ such that:

$$\min_{\theta} \; \beta_1 Emb_{Loss} + \beta_2 F_{Loss} + \beta_3 D1_{Loss}$$

- Train the extractor with parameters λ such that:

$$\min_{\lambda} \; \beta_4 E_{Loss} + \beta_5 D2_{Loss}$$

2.6 Training on Attack Simulations

To handle situations like intermediate attacks to remove the message and to check the robustness, a few popular attacks like gaussian blurring, rotation, cropping, and various noises were simulated. The mentioned attacks were simulated on the encoded images and tested the extractor on the attacked samples to produce undistorted spectrogram. Experimentally, the network is inherently robust towards the mentioned attacks, but there is always a scope for a new attack that can break the robustness. The remarkable feature of the network is that it can be trained to recover the message under any attacks. The procedure is similar to that of Algorithm 1, except by inputting the attacked encoded images to the model and train the extractor to recover the message.

2.7 Implementation Details

For the sake of simplicity, we chose 800 audio files of 2 s length those are sampled at 8 kHz. For the embedder, stacks of convolution layers with filters 16, 32, 64, 128, 256 and stride 2 was used instead of max pooling to avoid the losses due to the pooling layer. This has shown an improvement in the generator's loss. Also

the input dimensions were chosen as $128 \times 128 \times 3$. The **AE**'s encoder output is of size $2 \times 2 \times 256$ that is concatenated with the cover information of same size in the hiding network. The discriminators used were typical CNNs with an output layer of 1 dimension. In order to obtain a reduction of parameters, we avoided the extensive usage of fully connected layers. We had a total of 6 million parameters in the **Emb**, 2 million parameters in **AE**, 2.2 million parameters in **Ex** and, 1 million parameters in **D1** and **D2**. Adam optimizer with learning rate 0.0002 and momentum 0.5 was used as it is suitable for deep networks. The computational complexity in terms of training time for VoI-GAN with Div2K dataset is around 2 s for 1 epoch with GTX 1080Ti graphics processor.

3 Experiment

3.1 Dataset Details

For initial results, we have used the popular images in image processing society as referred in Table 1. We have used the DIV2K dataset [2] to train and evaluate our model. The dataset consists of 900 images, and we report the average PSNR, and SSIM on the test set. The training data comprises a total of 900 images where 800 are train set and the rest 100 are test images.

3.2 Without Attack Simulations

Apart from dataset analysis, we have also tested on various popular images as cover to check the generalization of the network and calculated the *PSNR* between the actual cover image and the encoded image. Shown in Table 1 are the comparison of different cover images their behavior in encoding the audio data and it is observed that the selection of cover image has not so significant role in encoding of data. *PSNR-CI* refers to the similarity between the encoded image and the cover image. As can be seen, there is no effect on the choice of the cover image. Our method works well for any cover image. On the Div2K dataset, we achieved embedding efficiency of **35.2 dB** PSNR. Extraction efficiency (measure of closeness between original and extracted spectrogram) of **31.44 dB** PSNR and finally structure similarity of **0.9828** between the cover and encoded image has been observed.

Table 1. Comparison on invisibility of message with different cover images.

Cover image	PSNR-CI (in dB)
Peppers	35.71
Mandrill	33.68
Tulips	36.08
Lena	33.95

3.3 After Attack Simulation

The model is quite efficient in hiding data in the cover image. For watermarking purposes, it should be capable of extracting the message from the encoded image even if it is attacked. Hence we test the efficiency of extraction on different audio files under different attacks. The extraction accuracy of the spectrogram of different files on attacks is shown in Table 2. As can be seen, the network is inherently robust to the considered attacks. We believe this inherent robustness comes from the deep architectures along with the autoencoder used to encode the spectrogram. From the extracted spectrogram, the audio is reconstructed using the Griffin-Lim method [6]. The mean opinion scores and *SSIM* of the reconstructed audio files are shown in Table 4. A common complaint from the M.O.S participants was the presence of a slight "hiss" sound in the background of the reconstructed audio. This can be avoided by adding a smoothing loss function in griffin-lim's method.

Table 2. Extraction efficiency of the spectrogram from attacked encoded Images in terms of PSNR (in dB). The last three columns corresponds to the types of noises.

Audio	Blur ($\sigma = 1$)	Rot90°	Rot10°	Crop 10%	S&P	Speckle	Gaussian
Female	55.26	26.86	51.89	58.17	54.80	54.32	54.12
Male	52.99	43.46	57.25	56.83	52.57	52.05	50.22
Car noise	52.76	24.95	51.62	53.90	52.96	53.24	52.92
White noise	51.99	20.81	57.73	52.04	51.33	51.79	51.40

To check the inherent robustness of the network, we gradually increased the blur variance, crop percentage, and rotation angles to plot *PSNR* and it is depicted in Fig. 3. We can further improve robustness by training the VoI-GAN with attacked samples as discussed in Sect. 2.4. However, the plot describes that the network is inherently robust to the popular attacks like blurring, crop, and rotation.

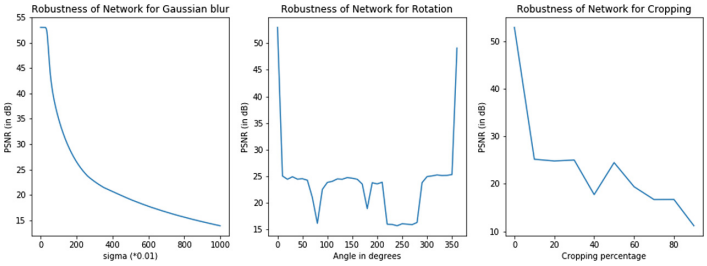

Fig. 3. Robustness of the models for the attacks before attack training. The extraction accuracy *PSNR* decreases with increase in blur, crop, and rotation till 180°. From 180° it is symmetry and hence the valley type of curve.

Since prior work in this area of hiding audio in images is a little scarce, we compared VoI-GAN with two other methods (Model 1 and Model 2) that we engineered over the time period in which we finally developed VoI-GAN. The architectural difference we had with Model 1 is that the extractor doesn't exist as a separate model. Instead, we trained the auto-encoder generator to embed and extract at the same time. This has reduced the number of parameters, however, this doesn't give the freedom to use multiple messages after training. The encoder takes the encoded image as input and is trained to output the message (spectrogram), while the entire generator takes the cover image as input and outputs similar image. Since encoder of the generator can extract the message, it is safe to say that the output of the generator has info about the message, making it "encoded" image. Hence the encoder of the generator is considered as the extractor. To prove that the existing image hiding techniques do not perform well on audio hiding in images, we propose Model 2 which is similarly built as Model 1, except we directly used the audio signal instead of the spectrogram and autoencoder base learning. Table 3 shows the *PSNR* and *SSIM* of the extracted audio files. We have ensured that Model 1 and Model 2 are similar to the prior work [10,12] by testing our Models on Div2K image dataset hiding. Model 2 directly hides the time-domain audio signal, while Model 1 hides the spectrogram. Even though Model 2 gave a decent *PSNR* and visually good extraction, the reconstructed audio was very bad. This is because of the fact that audio is highly non-stationary and can not be hidden directly by using prior methods.

Table 3. Comparison of audio extraction efficiency in terms of SSIM, PSNR (in db) and Mean Opinion Score between different models.

Model	SSIM	PSNR	Mean opinion score
Model 1	0.78	17.47	4.25
Model 2	0.81	21.42	0.00
VoI-GAN	0.998	53.45	4.53

Table 4. Audio Similarity in terms of SSIM and Mean opinion score between original and reconstructed audio file without any attack simulation on encoded Image

Audio file	SSIM	M.O.S
Female	0.998	4.53
Male	0.997	4.56
Car noise	0.992	4.49
White noise	0.989	4.43

4 Conclusion

The Audio over Image-GAN (VoI-GAN), to our knowledge, is the first end-to-end trainable adversarial model that can learn to embed audio signals into images.

The proposed method has shown robustness towards various attacks and hence can be used for watermarking images with audio as well. The network can be made more robust towards any attack by training, which is a remarkable feature. The proposed method is contributing to the field of Data Hiding by introducing a new and robust method to hide audio in images along with paving a research path for audio watermarking for an image. Due to inherent robustness, the model introduced in this work can be used to secretly transmit audio messages for strategic purposes. Methods to hide audio in images without deep learning would be very challenging and are hence not explored well. However, adopting deep learning ways may open new directions and find interesting applications in the field of data hiding. In the future, we would like to improve the architecture for multi-task training and replace the griffin-lim method with a deep network.

References

1. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In: ICLR (2016)
2. Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: dataset and study. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1122–1131 (2017)
3. Baluja, S.: Hiding images in plain sight: deep steganography. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems 30, pp. 2069–2079. Curran Associates, Inc. (2017)
4. Chen, B., Coatrieux, G., Chen, G., Sun, X., Coatrieux, J.L., Shu, H.: Full 4-D quaternion discrete Fourier transform based watermarking for color images. Digit. Sig. Process **28**(1), 106–119 (2014)
5. Das, C., Panigrahi, S., Sharma, V.K.: A novel blind robust image watermarking in DCT domain using inter-block coefficient correlation. Int. J. Electron. Commun. **68**(3), 244–253 (2014)
6. Griffin, D., Lim, J.: Signal estimation from modified short-time Fourier transform. IEEE Trans. Acoust. Speech Sig. Process. **32**(2), 236–243 (1984)
7. Feng, L.P., Zheng, L.B., Cao, P.: A DWT-DCT based blind watermarking algorithm for copyright protection. In: Proceedings of IEEE ICCIST, vol. 7, pp. 455–458 (2010)
8. Goodfellow, I., et al.: Generative adversarial nets. NIPS **27**, 2672–2680 (2013)
9. Ouyang, J., Coatrieux, G., Chen, B., Shu, H.: Color image watermarking based on quaternion Fourier transform and improved uniform log-polar mapping. Comput. Electr. Eng. **46**, 419–432 (2015)
10. Zhu, J., Kaplan, R., Johnson, J., Fei-Fei, L.: HiDDeN: hiding data with deep networks. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11219, pp. 682–697. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01267-0_40
11. Kandi, H., Mishra, D., Gorthi, S.R.S.: Exploring the learning capabilities of convolutional neural networks for robust image watermarking. Comput. Secur. **65**, 2506–2510 (2017)
12. Zhang, K.A., Cuesta-Infante, A., Xu, L., Veeramachaneni, K.: SteganoGAN: High capacity image steganography with GANs. arXiv preprint arXiv:1901.03892, January 2019

13. Mun, S.M., Nam, S.H., Jang, H.U., Kim, D., Lee, H.K.: A robust blind watermarking using convolutional neural network. arXiv preprint arXiv:1704.03248 (2017)
14. Su, Q., Niu, Y., Wang, Q., Sheng, G.: A blind color image watermarking based on DC component in the spatial domain. Optik **124**(23), 6255–6260 (2013)
15. Subakan, C., Koyejo, O., Smaragdis, P.: Learning the base distribution in implicit generative models. arXiv:1803.04357 (2018)
16. Gao, Y., Singh, R., Raj, B.: Voice impersonation using generative adversarial networks. In: Proceedings of ICASSP, pp. 2506–2510, April 2018