High-Capacity Convolutional Video Steganography with Temporal Residual Modeling

Xinyu Weng†, Yongzhi Li†, Lu Chi, Yadong Mu*
Peking University, Beijing 100080, China
{wengxy,yongzhili,chilu,myd}@pku.edu.cn

ABSTRACT

Steganography represents the art of unobtrusively concealing a secret message within some cover data. The key scope of this work is about high-capacity visual steganography techniques that hide a full-sized color video within another. We empirically validate that high-capacity image steganography model doesn't naturally extend to the video case for it completely ignores the temporal redundancy within consecutive video frames. Our work proposes a novel solution to this problem(*i.e.*, hiding a video into another video). The technical contributions are two-fold: first, motivated by the fact that the residual between two consecutive frames is highly-sparse, we propose to explicitly consider inter-frame residuals. Specifically, our model contains two branches, one of which is specially designed for hiding inter-frame residual into a cover video frame and the other hides the original secret frame. And then two decoders are devised, revealing residual or frame respectively. Secondly, we develop the model based on deep convolutional neural networks, which is the first of its kind in the literature of video steganography. In experiments, comprehensive evaluations are conducted to compare our model with classic steganography methods and pure high-capacity image steganography models. All results strongly suggest that the proposed model enjoys advantages over previous methods. We also carefully investigate our model's security to steganalyzer and the robustness to video compression.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision tasks**; • **Information systems** → *Information systems applications*;

KEYWORDS

Video Steganography, Deep Neural Networks, Temporal Modeling

ACM Reference Format:
Xinyu Weng, Yongzhi Li, Lu Chi, Yadong Mu. 2019. High-Capacity Convolutional Video Steganography with Temporal Residual Modeling. In *2019 International Conference on Multimedia Retrieval (ICMR'19), June 10–13, 2019, Ottawa, ON, Canada.* ACM, New York, NY, USA. 9 pages. DOI: https://doi.org/10.1145/3323873.3325011

ICMR'19, June 10–13, 2019, Ottawa, ON, Canada

ACM ISBN 978-1-4503-6765-3/19/06…$15.00
https://doi.org/10.1145/3323873.3325011

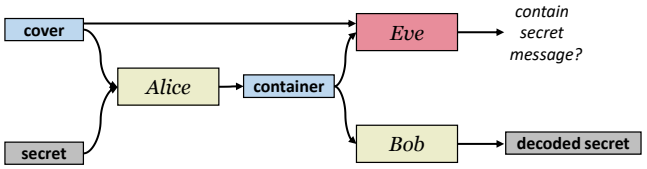

Figure 1: The full scheme of steganography. See main text for more explanation.

1 INTRODUCTION

The term steganography [2, 23, 24, 26] can date back to the 15th century, whose goal is to encode a secret message in some transport medium (called *cover* in this paper) and covertly communicate with a potential receiver who knows the decoding protocol. Essentially different from cryptography, steganography aims to hide the presence of secret communications, allowing only the target recipient to know. State differently, the covering medium can be publicly visible and yet only the target receiver can perceive the presence and decode the secret message. In practice, any steganography model should conceal a secret message by concurrently optimizing two criteria: minimizing the change of the covering medium that leads to suspect from an adversary, and reducing the residual between decoded secret message and its ground truth. The research on steganography has practical implications. For example, a number of nefarious applications of steganography techniques are known, such as hiding commands that coordinate criminal activities through images posted on social media websites. In the industry of digital publishing, a common tactic to claiming authorship without compromising the integrity of the digital content is to embed digital watermarks. For other brief introduction to steganography, one can refer to [1, 29, 31, 35].

Let us first explain the process of a typical steganography system, which is shown in Figure 1. In classic steganography, the process involves three parties: Alice, Bob and Eve. Alice first conceals a *secret* message into a *cover* to obtain a *container* message, then sends the container to Bob. Eve is an adversary (the *steganalyzer*) to both Alice and Bob. His goal is to judge whether a message he observed is steganographic or not. But he is not requested to decode the hidden secret. In this scheme, we say Alice performs perfectly if she ensures: 1) Bob receives the container and recovers secret at high accuracy using a decoding protocol; and 2) Eve has exactly 50% chance of correctly judging a container or cover. It is similar to the expectation in adversarial training [11, 16]. To accomplish both goals, the container should not deviate from the original cover too much, avoiding that abnormal pattern appears and is detected by

† denotes equal contribution. * is the corresponding author.

Eve. Meanwhile, it should also be in a good shape to be accurately deciphered by the decoder model at Bob's hand.

Hiding messages in an image has been a long-standing research task of salient practical interest [8, 13, 21, 25, 38]. One can gauge the amount of concealed information through bits-per-pixel (bpp), namely the amortized bits hidden at each pixel in the cover image. Traditional image steganography can only handle very little secret information (usually lower than 1 bpp) [12]. While a recent research trend is hiding high bpp secret as exemplified in [3], which encodes a full-sized color image into another same-sized image(high-capacity image steganography). This represents a highly challenging task since it pursues a bpp level 24 (*i.e.*, each pixel in the cover hides a complete RGB color). Figure 2 illustrates a typical results calculated from a high-capacity image steganography model. The steganography model can hardly accomplish both of Alice's two goals in the container. As artifacts can often be observed in container, making it easily detected by an adversary.

In this work, our major focus is video steganography. The task aims to hide a full-sized video clip into another. Considering the increasing popularity of video data across the Internet, the research of video steganography, though currently rarely found in the literature, represents a nascent research topic of key practical implications. It is naturally regarded that high-capacity image steganography model can be readily used to solve the video steganography problem, by pairing frames in cover / secret videos and feeding them into an image model. We argue that this tactic is not optimal, because it does not fully consider the temporal redundancy within consecutive video frames. Our work proposes a novel solution to video steganography. Briefly speaking, the technical contributions are two-fold:

First, the residuals between two consecutive frames are highly sparse. Critically, compared with hiding frame into another frame, hiding such sparse residual in another video frame defines a much easier task. Motivated by this fact, instead of blindly applying image model on all frames, we propose to split frames into two sub-sets: *reference frames* and *residual frames*. Each residual frame is obtained by differencing with specific reference frame. Correspondingly, our model contains two branches at both the encoding and decoding stages, tackling either type of frames respectively. We empirically validate this treatment can significantly boost the container's perceptual quality and increase the possibility of fooling an adversary.

Secondly, our model is fully based on deep convolutional neural networks, which is the first of its kind in video steganography. Specifically, our deep video steganography model consists of two H-networks for hiding references or residuals, and two R-networks for revealing the secret video. The full model is trained without any human annotations and network parameters are optimized from scratch. In experiments, comprehensive evaluations are conducted to validate the powerful modeling of deep networks. We also carefully design ablation investigation to find key factors in our deep video steganography model.

The remainder of this paper is organized as following: We first briefly review the related work in Section 2. Section 3 details the proposed two-branch deep neural networks for the video steganography task. All experimental evaluations and in-depth analysis are found in Section 4. Finally, Section 5 concludes this work and points out several future research directions.

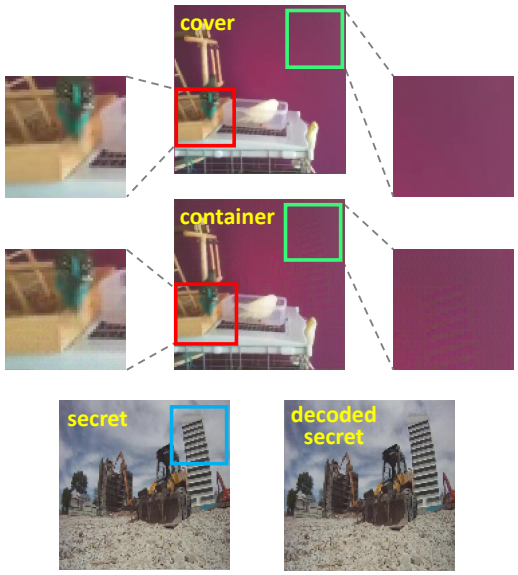

Figure 2: Exemplar results generated by a high-capacity image steganography model [3]. The role of each image is depicted in bold yellow text located in the top-left of each image. To depict how container image deviates from the original cover image, we choose two local patches and contrast them for these two images. Indeed, for the local patch delimited by the green box, from the container image one can observe the ghost image of specific building in the secret image (in blue box). Better viewing after enlarging.

2 RELATED WORK

Least Significant Bit (LSB) [5, 14, 34, 42] is a classic steganographic algorithm. In digital images, each pixel in an image is comprised of three bytes (*i.e.*, 8 binary bits), representing the RGB chromatic values respectively. The *n*bit-LSB algorithm replaces the least n significant bits of the cover image by n most significant bits of the secret image. For each byte, the significant bits dominate the color values. This way, the chromatic variation of the container image (altered cover) is minimized. Revealing the concealed secret image can be simply accomplished by reading the n least significant bits and performing bit shift. Despite that its distortion is not often visually observable, LSB is unfortunately highly vulnerable to steganalysis [15, 30, 33] - statistical analysis can easily detect the pattern of altered pixels. Recent works have been devoted to more sophisticated methods that preserve the image statistics or design special distortion functions [18, 19, 32, 39, 48, 49].

To overcome the drawbacks of LSB, the variant HRVSS [10] and [36] exploits special biological trait of human eyes for hiding a grey image in a color image. Several other works utilize bit plane complexity segmentation in either spatial or transform domain [28, 44, 47]. Other algorithms [4, 27, 37, 53] embed secret in DCT (Direct Cosine Transformation) domain by changing DCT coefficients. As many coefficients are equal to zero, changing too many zeros to non-zero values will cause large distortion in container. It explains that few bits can be embedded in DCT domain than spatial domain [6, 9, 45].

Recently, several deep learning based steganography methods are developed to encode text message in images, such as the works

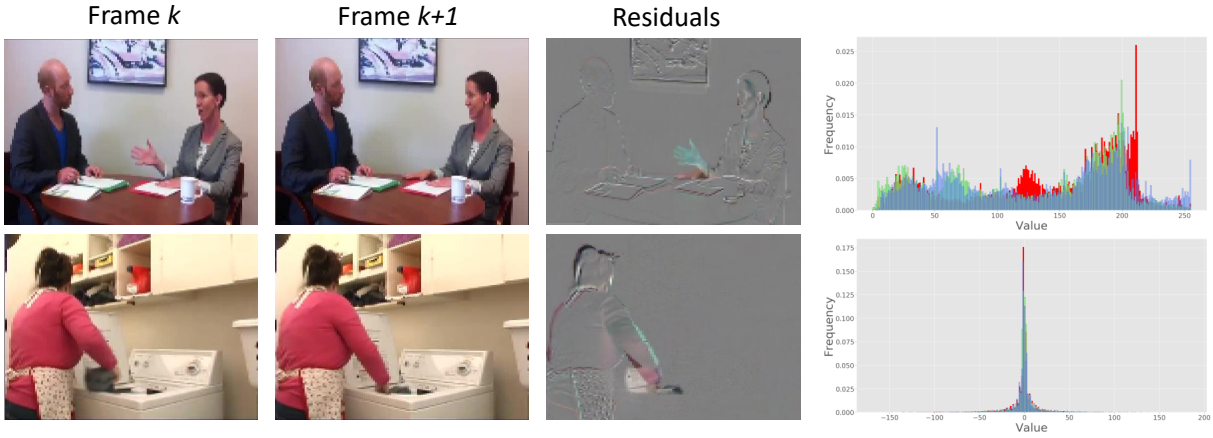

Figure 3: Examples of video frames and inter-frame residuals. The column *residuals* represent the per-pixel difference between frame k and $k+1$. The righmost column shows the distribution of RGB values (top) and residual values (bottom) for the first frame pair (top row).

in [3, 54]. Early works [20, 40] mostly focused on the decoding step (such as determining which bits to extract from the container images) to elevate accuracies. Other works investigate efficiency of employing deep learning on steganalysis such as [41, 43, 50, 52]. Both of [3, 17] build the whole system based on deep networks, including encoding (hiding) and decoding (revealing) networks. Prior quantitative evaluations strongly corroborate the superior modeling ability of deep networks in image steganography. However, to our best knowledge, there is no prior work that explores deep networks for the hiding-video-in-video setting.

3 THE PROPOSED MODEL

Figure 3 illustrates some motivating fact to our video steganography model. As seen, the residual values between consecutive video frames are dominated by near-zero values. Hiding such high-sparse data into a cover frame intuitively requires less effort compared with a full-colored secret frame, since hiding a zero value is trivial. This way, the cover image tends to be less altered, which potentially increases the chance of fooling an adversary. Using residuals as the secret message instead can ease Alice's job (or the encoding model) in Figure 1 and meanwhile does not make Bob's task harder. However, to operate on residuals, there are two challenges that we should concern: how to determine encoding the original video frame or its residual with respect to the previous frame? And at the decoding stage, how the decoder knows the received image conceals a full-colored frame or a residual array?

To address above issues, we categorize all secret frames to be either *reference frame* or *residual frame*. Correspondingly, we propose to use two separate encoding / decoding networks for tackling different type of frames. The architecture of our proposed system is shown in Figure 4. The system is comprised of six computational steps.

3.1 Computational Pipeline

Step-1: Reference/Residual Frame Labeling : We adopt a simple thresholding approach for labeling a frame to be reference or residual type. Specifically, the first frame in a video is surely labeled as reference. The following frames in the same video sequentially calculate their *averaged pixel-wise discrepancy* (APD)[1] with respect to the first frame. Once the APD score of any frame exceeds some pre-specified threshold, it will be set as a new reference and used to calibrate all following frames. The procedure proceeds until all frames are labeled.

Step-2: Hiding Secret: This step does Alice's job in Figure 1. The key difference of our method to others is a divide-and-conquer scheme. Note that in Figure 4 two hiding networks are devised, referred to as *Reference H-net* or *Residual H-net* respectively. Concatenated with cover frame F_{cov}, each secret frame F_{sec} is fed into the corresponding H-net by their label and the output is container frame F_{con}.

Step-3: Video Codec Simulation: In practical applications, Alice may compress a video (*e.g.*, in MP4 or MPEG format) before sending it to Bob. A video that goes through the video encoding / decoding process can largely deviate from its original version. When deep networks are utilized, small perturbation of container video can be gradually enlarged at later neural layers and may cause a large deviation in Bob's revealed video. To mitigate this problem, we add a Codec Simulation Layer (CSL) for simulating the video codec process. For lossless video compression, CSL is simply an identity mapping that does nothing on its input. For lossy compression, we design the layer by investigating some statistics of video pixels. More details are deferred to the Experiment Section.

Step-4: Revealing Secret: It does Bob's job in Figure 1. The input is merely the container frame $F_{con'}$ after codec, and the output (we call it *revealed secret* F_{rev}) is another frame which is desired to be exactly the secret in the perfect case.

Similar to H-nets, two R-nets (*Reference R-net* or *Residual R-net*) are introduced to reveal the frame or residual secret. However, unlike the hiding stage, Bob strictly has no access to the cover or secret, which implies that frame labels are missing. State differently, the receiver is not aware of which R-net is the optimal handler. We

[1] For two RGB frames, we calculate pixel-wise absolute difference and take the average for R, G, and B-channel respectively. The APD score is defined as the average value across 3 channels.

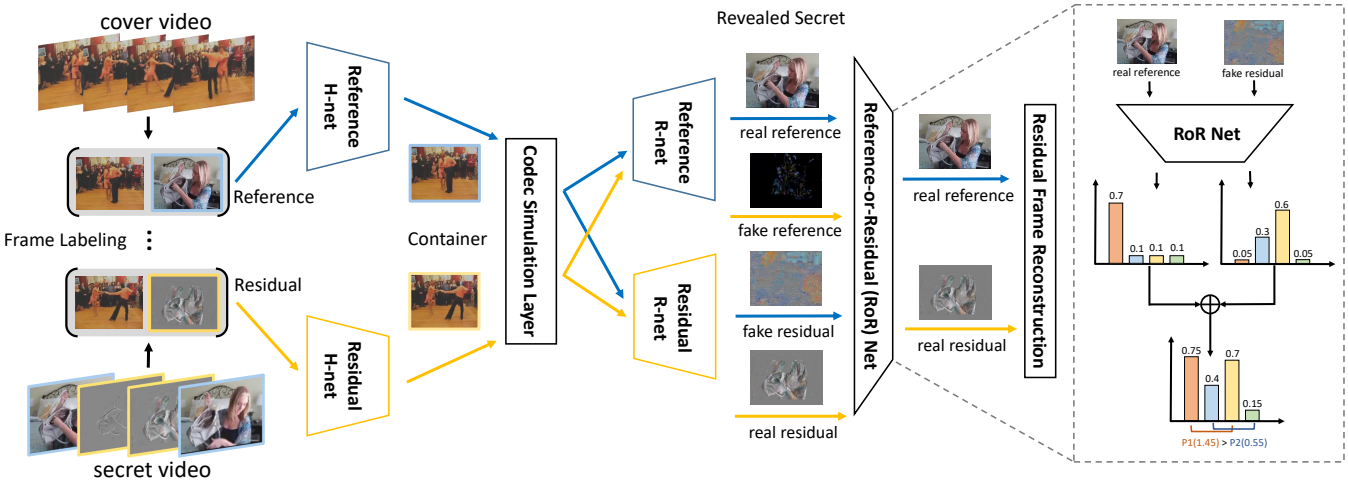

Figure 4: The computational pipeline of our proposed video steganography model. Each container passes through two R-nets respectively to get two revealed messages. The subfigure on the right shows the classification mechanism of the RoR-net.

postpone this decision to the next step. The container frame will be sent to both R-nets and obtained two revealed secret frames.

Step-5: Reference-or-Residual Classification: Our proposed temporal residual modeling raises new challenges to the classic scheme as depicted in Figure 1 - Bob receives two copies of revealed secret messages in Step-4, from Reference R-net or Residual R-net respectively. Clearly, only one of the secret message is true. Bob needs to pick out the real message. In fact, we can exhaustively enumerate all possible messages: the real reference and fake residual (container with a true reference secret gets through Reference and Residual R-nets respectively), real residual or fake reference (similar to above, but containers now carry residuals), totalling four valid cases. Therefore, we formulate it as a four-way classification problem. As seen in Figure 4, a Reference-or-Residual (RoR) Net is devised for judging an input revealed message.

Step-6: Residual Frame Reconstruction: This step is optional if Step 5 judges a message as real reference. However, for a residual frame, it is not visually understandable per se. One need to add revealed residuals to the correct reference frame for obtaining the concealed video frame. Since we always process video frames in temporal order, we can record the latest reference frame for reconstructing residuals. Due to the addition operation, an unavoidable problem is such reconstruction scheme will introduce two parts of errors to reconstructed residual frame (from the corresponding reconstructed reference and the residual itself). To tackle this problem, we assume Alice has access to the protocol of revealing process. When labeling reference and residual frames in step-1, she first sends the frame to Reference H-net and R-net to get the revealed reference secret. Instead of comparing current secret frame with latest secret reference, the residual frame is labeled by calculating the APD between current secret and latest revealed reference secret.

3.2 Hiding / Revealing Networks

In our proposed system, each pair of H-net / R-net for hiding / revealing specific type of frame is jointly trained before the RoR net. Each H-nets take the concatenation of cover frame F_{cov} and corresponding secret frame F_{sec} as input and output container frame F_{con}. In practice, we choose the U-net model [7, 22] for both H-nets. The network specifications are found in Table 1. We use the following loss to measure distortion between cover and container

$$\mathcal{L}_H(F_{cov}, F_{con}; H_\theta) = ||F_{cov} - F_{con}||_F^2, \tag{1}$$

where $||\cdot||_F$ is the Frobenius norm.

We let R-net have a mainframe of five convolutional layers, each of which is paired with BN layer and LeakyReLU. The input of R-net is container frame $F_{con'}$ after codec and the output is revealed secret $F_{revealed}$. The specification of R-nets is found in Table 2. The R-net models are trained to minimize the discrepancy between the secret and its revealed version:

$$\mathcal{L}_R(F_{sec}, F_{rev}; R_\theta) = ||F_{sec} - F_{rev}||_F^2. \tag{2}$$

We define overall loss function for learning H-nets / R-nets as $\mathcal{L}_{sum} = \mathcal{L}_H + \lambda\mathcal{L}_R$. Here constant λ is used to balance the perceptual performance of container and revealed secret. For all experiments, λ is set as 0.75. It should be clarified that all nets do not share any parameter.

3.3 Reference-or-Residual (RoR) Network

As stated earlier, to categorize the revealed message we train a four-class CNN Reference-or-Residual (RoR) classifier. In practice, we use the trained Reference H/R-nets and Residual H/R-nets to t training data, and use these data to train the RoR network. The architecture of RoR-net is similar to R-net except the network head, which is a linear fully-connected layer followed by a softmax layer. Given an input image, the softmax eventually returns a 4-d probabilistic vector that categorizes the revealed information. For learning the RoR net, we adopt the standard cross-entropy loss to enforce label consistency.

On the testing set, an accuracy of 99.9625% was achieved, which is nearly perfect yet the RoR-net is still fooled by some hard samples. To attack this issue, we propose an improved judgment method.

Table 1: Architecture of both Reference Hiding network and Residual Hiding network. There is a batch normalization layer (BN) and a Leaky Rectified Linear Unit (LeakyReLU) after each convolution layer. And there is a BN and a Rectified Linear Unit (ReLU) after each deconvolution layer except for the last one. The last deconvolution layer is followed by a *Sigmoid* function.

Index	Type	Kernel	Stride	Padding	Input	Out	Concatenation
1	Conv2d.	4×4	2	1	6	64	N/A
2	Conv2d.	4×4	2	1	64	128	N/A
3	Conv2d.	4×4	2	1	128	256	N/A
4	Conv2d.	4×4	2	1	256	512	N/A
5	Conv2d.	4×4	2	1	512	512	N/A
6	Conv2d.	4×4	2	1	512	512	N/A
7	Conv2d.	4×4	2	1	512	512	N/A
8	deConv2d.	4×4	2	1	512	512	N/A
9	deConv2d.	4×4	2	1	1024	512	concat with layer #6
10	deConv2d.	4×4	2	1	1024	512	concat with layer #5
11	deConv2d.	4×4	2	1	1024	256	concat with layer #4
12	deConv2d.	4×4	2	1	512	128	concat with layer #3
13	deConv2d.	4×4	2	1	256	64	concat with layer #2
14	deConv2d.	4×4	2	1	128	3	concat with layer #1

Table 2: Architecture of both Reference Reveal network and Residual Reveal network. Each layer has a 3×3 convolution. There is a batch normalization layer (BN) and a Rectified Linear Unit (ReLU) after each convolution layer except for the last one. The output convolution layer is followed by a *Sigmoid* function.

Index	Type	Kernel	Stride	Padding	Input	Out
1	Conv2d.	3×3	1	1	3	50
2	Conv2d.	3×3	1	1	100	50
3	Conv2d.	3×3	1	1	100	50
4	Conv2d.	3×3	1	1	100	50
5	Conv2d.	3×3	1	1	100	50
5	Conv2d.	1×1	1	0	100	3

Given a specific type of container frame, the Reference and Residual R-nets will output two revealed messages. Then RoR-net will output two 4-d probabilistic vectors. Because there are only two combinations of reference and residual values, i.e. real reference and fake residual (generated by container with reference frame) or fake reference and real residual (generated by container carrying residual frame), we calculate a final score vector by executing element-wise addition of the two probabilistic vectors. After that, we add the score of real reference and score of fake residual as P1. The score of fake reference and the score of real residual was added up as P2. If P1 is larger than P2, we suppose that this container conceals reference information, otherwise it hides residuals. The subfigure of Figure 4 shows a classification process example of a container with a true reference frame. This simple scheme brings a 100% accuracy on the test set. It is worth noting that this accuracy is obtained on a set of 24,000 samples, so though small($< 5e-5$), the possibility of misclassification of references and residuals exists. If a frame is misclassified unfortunately, the successive frames will be affected until the next reference is correctly classified. This infrequent error can be reduced by choosing a smaller threshold (narrowing the interval of reference frames).

4 EXPERIMENTS

4.1 Dataset Description and Experimental Setting

There is no available benchmark used for video steganography research. We therefore construct a new benchmark as follows: TRECVID Multimedia event detection (MED)[2] is a yearly competition about retrieving specific semantic events (such as “birthday party" or “parkour") from a huge pool of videos. The MED 2017 video corpus consists of more than 0.3 Million videos with high-quality annotation. Since our task is essentially unsupervised, we ignore the video semantic labels and randomly sample 12,000 videos from the whole set. For each video, a 2-second clip is randomly cropped and 24 frames are extracted using the tool of FFMPEG. We generate a data split of training / validation / testing subsets, with 10,000, 1,000, and 1,000 video clips respectively.

We get the splitting threshold 30 by calculating mean APD between the twelfth frame and the first frame on all training data and generate 43,610 reference frames and 196,840 residuals. Videos are randomly drawn to form the (cover, secret)-pair. The Reference H-net is trained using all reference frames, and Residual H-net utilizes the residuals. All decoded messages collectively train the four-way

[2]http://www-nlpir.nist.gov/projects/tv2017/Tasks/med/

Figure 5: Hiding results using our video model. Left pair of each set: original cover and secret. Center pair: cover frame embedded with the secret frame (container) and the revealed secret frame. Right pair: Residual errors for container and secret (enhanced 5x). Secret frames in odd and even rows are reference / residual frames respectively. The results are achieved by setting CSL to identity-mapping layer.

RoR net. All frames are resized to 256×256 pixels before sending to networks. We tune the network parameter using Adam with standard parameters and use an initial learning rate of 0.001 that is decayed by a factor of 10 each time the validation loss plateaus after 5 epochs. The best model on the validation set is kept as the final model.

4.2 Empirical Evaluation and Analysis

Figure 5 shows the steganography results on selected videos. For each video, we show both the results of Reference H/R-nets and Residual H/R-nets. By investigating the residuals between container-cover and secret-revealed secret pairs as in Figure 5, one can observe that the container frames still look visually natural and the residual error is smaller when hiding a residual frame. Since no existing work exploring hiding video in another video, we choose four best-known steganography methods that have comparable capacity for information embedding with ours, including 4bit-LSB, HRVSS [10], Baluja [3] and HiDDeN [54]. HRVSS uses an improved LSB strategy to hide 8 bits of one gray image in three channels of another RGB image. Although the input secret is a color image, it can only reveal its gray version. As HiDDeN is not specially designed for high-capacity steganography, we reimplement its input and output layer to ensure the consistence of experiment settings. In Table 3, we report several performance measures on visual similarity and quality loss between cover-container pair and secret-revealed secret pair respectively, including APD, RMSE, PSNR (Peak Signal-to-Noise Ratio), SSIM (Structural Similarity Index) and VIF (Visual Information Fidelity) [46]. We also perform visual comparison with other methods in Figure 6 and clear superiority goes to our model. Both 4bit-LSB and Baluja output containers and revealed secrets with obvious textures. Undesirable color bias phenomenon can be found in the result frames of HiDDeN. It is seen that our full model enjoys few distortions for both the container and revealed secret frame. Our model yields optimal performance in both image purity and color fidelity. It is clarified that all the results are achieved under the setting of lossless transmission and the CSL in our model is set to identity-mapping layer.

4.3 Investigation on Adversarial Learning

In steganography, a prominent goal is to fool the adversary, Eve in Figure 1. An interesting problem to us is: after collecting how many labeled cover / container data, the adversary will become accurate enough to detect the presence of secret message? Without loss of generality, we assume the adversary uses a 6-layer CNN for learning a binary classification from labeled data. We investigate both LSB and our video model, as shown in Figure 7. Interestingly, both methods tend to have zero probability of fooling the adversary after about 2,000 labeled data are leaked.

To increase the resistance to the adversary, we explore an idea of adding an adversarial learning sub-model, similar to [17]. Specifically, besides H/R-nets we incorporate an adversarial discriminator (we assume it has a common CNN architecture). It can receive frames from the cover and container to make a judgment. If the discriminator cannot distinguish a cover or container, it means that the container generated by H-nets is able to fool this discriminator. We adopt the classic GAN [16] loss for this discriminator. In one of

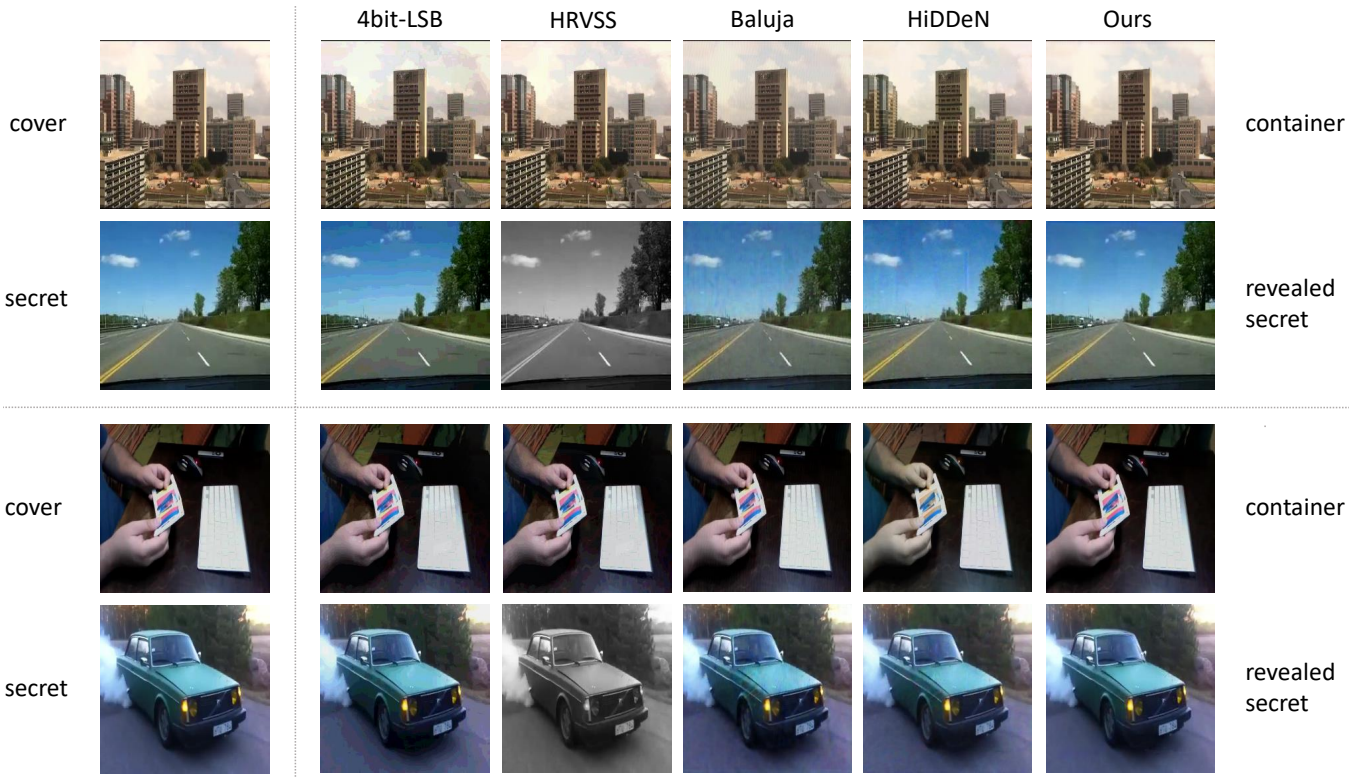

Figure 6: Comparison of perceptual quality with 4bit-LSB, HRVSS [10], Baluja [3] and HiDDeN [54]. Left column: Cover and secret frames. Right columns: containers and revealed secrets. All results are achieved under the setting of lossless transmission. Our model achieves better color fidelity and minor residual error than others. Better viewing after enlarging.

Table 3: Comparison of quality measures on cover-container pair and secret-decoded secret pair under the setting of lossless transmission. ↑ denotes higher is better, and vice versa.

method	cover-container pair					secret-revealed secret pair				
	PSNR↑	SSIM↑	VIF↑	RMSE↓	APD↓	PSNR↑	SSIM↑	VIF↑	RMSE↓	APD↓
4bit-LSB	31.88	0.6287	0.6145	6.59	5.51	29.41	0.6550	0.6405	8.73	7.29
HRVSS [10]	39.95	0.7735	0.8038	2.83	2.18	24.93	**0.8702**	0.3634	20.12	11.79
Baluja [3]	38.97	0.7796	0.7759	3.05	2.17	33.91	0.7501	0.6649	5.39	3.92
HiDDeN [54]	32.13	0.8267	0.6265	7.24	5.31	34.19	0.7462	0.6850	5.24	3.80
ours	**40.62**	**0.8466**	**0.8286**	**2.50**	**1.68**	**40.76**	0.8542	**0.8368**	**2.50**	**1.66**

Table 4: Comparison of quality measures with/without CSL under the setting of lossy transmission.

method	cover-container pair					secret-decoded secret pair				
	PSNR↑	SSIM↑	VIF↑	RMSE↓	APD↓	PSNR↑	SSIM↑	VIF↑	RMSE↓	APD↓
train without CSL	40.62	0.8466	0.8286	2.50	1.68	24.75	0.4614	0.3617	19.13	10.82
train with CSL	36.51	0.7533	0.7127	3.87	2.82	33.43	0.6714	0.6023	5.54	4.18

our experiments, we combine this GAN loss with H/R losses, modulated via a hyper-parameter (set to 3e-4 empirically), and re-train the video model. The APD of the re-trained H/R-nets are 2.81 and 3.32, respectively, which are slightly worse than the original video model in Table 3. This implies that the new loss sacrifices APD in H/R-nets for lower detectability by the adversary.

After re-training, we plot the curve of the new model in Figure 7 (denoted as "w/ GAN"). The apparent gap between the curves of

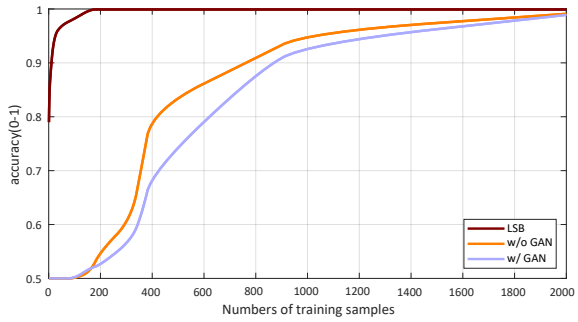

Figure 7: Investigation on adversarial learning. The curves indicate how many leaked training samples are needed for an adversary to distinguish a container from a cover, which reflects the anti-steganalysis ability of a model.

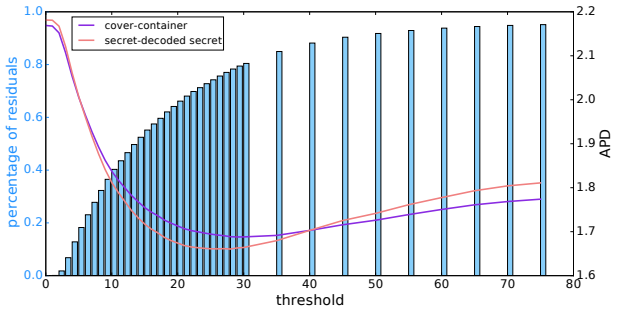

Figure 8: Comparison with different splitting threshold on test set. The light blue bar denotes percentage of residual frames generated under the certain threshold.

original model (denoted as "w/o GAN") and new model indicates that the latter is enhanced for anti-steganalysis. For example, for the original model, leaking 400 training pairs can enable the attacker to correctly distinguish 80% testing samples. While for the adversarially-trained new model, to achieve this accuracy, more than 600 pairs are required. This experiment serves a strong evidence that incorporating a GAN-style adversarial discriminator can lead to a more steganalysis-secure message embedding. It is also noted that, for LSB the adversary can easily perform shift operations on covers and containers to distinguish, making it less secure.

4.4 Codec Simulation

Let us detail the design of Codec Simulation Layer (CSL). Since video codec is generally non-differentiable, we introduce some approximation for ensuring gradient's back propagation through this layer. Specifically, We use `image2` demuxer to sythesize mp4 video file from container frames via FFMPEG (parameter `qscale` is set to 0 and `vb` is 100M), and then extract frames from the compressed video. After comparing the frames before and after codec, we find that the variation of each pixel approximately obeys the lognormal distribution, which motivates us to utilize random noise generator for simulating codec. CSL is initialized as identity mapping. After a few epoches of training, CSL switches to draw random noises from the lognormal distribution independently for each pixel and adds the noise to the original pixel value. This can be regarded as a tractable simulation of codec.

Table 4 shows that under the setting of lossy transmission, the results of training with/without CSL are quite different. In the first experiment of Table 4, we use the same H-net and R-net as those in experiments of Table 3, and the performance of the container is not affected. The container is then compressed and decompressed to get the container'. As the parameters of R-net trained on container are not applicable to container', the visual performance of revealed secret is surely very poor (with undesirable light spots) and all measures of secret-revealed secret pair get worse sharply. After fine-tuning H-net and R-net with CSL in an end-to-end manner, the parameters of our model can adapt to the effects of video codec and restored secrets are consistently better on all measures. The randomness introduced by CSL slightly reduces the visual performance of the container, but greatly enhances the performance of the revealed secret. This proves the addition of CSL successfully mitigates the problem of poor performance of revealed secret caused by video codec in practical use at a small price.

4.5 The Choice of Threshold

We adopt a thresholding scheme to split reference and residual frames. However, choosing a proper threshold is non-trivial. When selecting different threshold, the APDs between cover-container pair and secret-revealed secret pair are presented in Figure 8. Enlarging the threshold will generate more residuals with more information, making the residual branch harder to reveal the residual secret and degrading the final revealed secret. If we set a smaller threshold, there will be more reference frames, making the video model quickly converge to the image steganography. In practical use, one may adjust the threshold for different needs and application scenarios. We consider the threshold is a trade-off of quality between the container and revealed secret. For example, it is possible to enlarge the threshold when the security of container is more important than quality of decoded secret. In our test stages, the threshold is set to 30 to ensure the performance of both container and decoded secret.

5 CONCLUDING REMARKS

We present a novel deep neural network for the task of high-capacity video steganography. To fully utilize the sparse property of inter-frame differences, we develop a temporal residual modeling technique, separately treating reference and residual frames during generating steganographic videos. We also take into consideration the effect of video codec process in lossy transmission. Comprehensive evaluations and studies show the superiority of our method. The future work shall include the exploration of more sophisticated deep models, such as C3D [51], which may better model temporal relationship between frames.

ACKNOWLEDGMENTS

This work is supported by Beijing Municipal Commission of Science and Technology under Grant 181100008918005, National Natural Science Foundation of China (NSFC) under Grant 61772037 and a start-up grant from Peking University.

REFERENCES

[1] George Abboud, Jeffrey S. Marean, and Roman V. Yampolskiy. 2010. Steganography and Visual Cryptography in Computer Forensics. In *SADFE*.

[2] Ross J Anderson and Fabien AP Petitcolas. 1998. On the limits of steganography. *IEEE Journal on selected areas in communications* 16, 4 (1998), 474–481.

[3] Shumeet Baluja. 2017. Hiding Images in Plain Sight: Deep Steganography. In *NIPS*.

[4] J. J. Chae and B. S. Manjunath. 1999. Data hiding in video. In *ICIP*.

[5] Rajarathnam Chandramouli and Nasir Memon. 2001. Analysis of LSB based image steganography techniques. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, Vol. 3. IEEE, 1019–1022.

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

[7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. 2016. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *CoRR* abs/1606.00915 (2016).

[8] Po-Yueh Chen, Hung-Ju Lin, et al. 2006. A DWT based approach for image steganography. *International Journal of Applied Science and Engineering* 4, 3 (2006), 275–290.

[9] Sahar A. El-Rahman. 2018. A comparative analysis of image steganography based on DCT algorithm and steganography tool to hide nuclear reactors confidential information. *Computers & Electrical Engineering* 70 (2018), 380–399.

[10] Mohamed Elsadig Eltahir, Miss Laiha Mat Kiah, Bilal Bahaa, and A Zaidan. 2009. High Rate Video Streaming Steganography. In *ICIME*.

[11] Gamaleldin F. Elsayed, Shreya Shankar, Brian Cheung, Nicolas Papernot, Alex Kurakin, Ian J. Goodfellow, and Jascha Sohl-Dickstein. 2018. Adversarial Examples that Fool both Human and Computer Vision. *CoRR* abs/1802.08195 (2018).

[12] Jessica Fridrich and Miroslav Goljan. 2002. Practical Steganalysis of Digital Images - State of the Art. (2002).

[13] Jessica Fridrich and Miroslav Goljan. 2003. Digital image steganography using stochastic modulation. In *Security and Watermarking of Multimedia Contents V*, Vol. 5020. International Society for Optics and Photonics, 191–203.

[14] Jessica Fridrich, Miroslav Goljan, and Rui Du. 2001. Detecting LSB steganography in color, and gray-scale images. *IEEE multimedia* 8, 4 (2001), 22–28.

[15] Jessica J. Fridrich, Miroslav Goljan, and Rui Du. 2001. Detecting LSB Steganography in Color and Gray-Scale Images. *IEEE MultiMedia* 8, 4 (2001), 22–28.

[16] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *NIPS*.

[17] Jamie Hayes and George Danezis. 2017. Generating steganographic images via adversarial training. In *NIPS*.

[18] Vojtech Holub and Jessica J. Fridrich. 2012. Designing steganographic distortion using directional filters. In *WIFS*.

[19] Vojtech Holub, Jessica J. Fridrich, and Tomás Denemark. 2014. Universal distortion function for steganography in an arbitrary domain. *EURASIP J. Information Security* 2014 (2014), 1.

[20] Sabah Husien and Haitham Badi. 2015. Artificial neural network for steganography. *Neural Computing and Applications* 26, 1 (2015), 111–116.

[21] Saiful Islam, Mangat R Modi, and Phalguni Gupta. 2014. Edge-based image steganography. *EURASIP Journal on Information Security* 2014, 1 (2014), 8.

[22] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. In *CVPR*.

[23] Neil F Johnson, Zoran Duric, and Sushil Jajodia. 2001. *Information Hiding: Steganography and Watermarking-Attacks and Countermeasures: Steganography and Watermarking: Attacks and Countermeasures*. Vol. 1. Springer Science & Business Media.

[24] Neil F Johnson and Sushil Jajodia. 1998. Exploring steganography: Seeing the unseen. *Computer* 31, 2 (1998).

[25] SM Masud Karim, Md Saifur Rahman, and Md Ismail Hossain. 2011. A new approach for LSB based image steganography using secret key. In *Computer and Information Technology (ICCIT), 2011 14th International Conference on*. IEEE, 286–291.

[26] Stefan Katzenbeisser and Fabien AP Petitcolas. 2002. Defining security in steganographic systems. In *Security and Watermarking of Multimedia Contents IV*, Vol. 4675. International Society for Optics and Photonics, 50–57.

[27] Blossom Kaur, Amandeep Kaur, and Jasdeep Singh. 2011. Steganographic approach for hiding image in DCT domain. *International Journal of Advances in Engineering & Technology* 1, 3 (2011), 72.

[28] Eiji Kawaguchi and Richard O Eason. 1999. Principles and applications of BPCS steganography. In *Multimedia Systems and Applications*, Vol. 3528.

[29] Gary C. Kessler and Chet Hosmer. 2011. An Overview of Steganography. *Advances in Computers* 83 (2011), 51–107. https://doi.org/10.1016/B978-0-12-385510-7.00002-3

[30] Daniel Lerch-Hostalot and David Megías. 2016. Unsupervised steganalysis based on artificial training sets. *Eng. Appl. of AI* 50 (2016), 45–59.

[31] Bin Li, Shunquan Tan, Ming Wang, and Jiwu Huang. 2014. Investigation on Cost Assignment in Spatial Image Steganography. *IEEE Trans. Information Forensics and Security* 9, 8 (2014), 1264–1277.

[32] Min Long and Fenfang Li. 2018. A Formula Adaptive Pixel Pair Matching Steganography Algorithm. *Adv. in MM* 2018 (2018), 7682098:1–7682098:8.

[33] Weiqi Luo, Fangjun Huang, and Jiwu Huang. 2010. Edge adaptive image steganography based on LSB matching revisited. *IEEE Transactions on information forensics and security* 5, 2 (2010), 201–214.

[34] Jarno Mielikäinen. 2006. LSB matching revisited. *IEEE Signal Process. Lett.* 13, 5 (2006), 285–287.

[35] T. Morkel, Jan H. P. Eloff, and Martin S. Olivier. 2005. An overview of image steganography. In *ISSA*.

[36] Khan Muhammad, Muhammad Sajjad, Irfan Mehmood, Seungmin Rho, and Sung Wook Baik. 2018. Image steganography using uncorrelated color space and its application for security of visual contents in online social networks. *Future Generation Comp. Syst.* 86 (2018), 951–960.

[37] Amitava Nag, Sushanta Biswas, Debasree Sarkar, and Partha Pratim Sarkar. 2010. A novel technique for image steganography based on Block-DCT and Huffman Encoding. *arXiv preprint arXiv:1006.1186* (2010).

[38] Mohammad Tanvir Parvez and Adnan Abdul-Aziz Gutub. 2008. RGB intensity based variable-bits image steganography. In *2008 IEEE Asia-Pacific Services Computing Conference*. IEEE, 1322–1327.

[39] Tomás Pevný, Tomás Filler, and Patrick Bas. 2010. Using High-Dimensional Image Models to Perform Highly Undetectable Steganography. In *Information Hiding*.

[40] Lionel Pibre, Jérôme Pasquet, Dino Ienco, and Marc Chaumont. 2015. Deep Learning for steganalysis is better than a Rich Model with an Ensemble Classifier, and is natively robust to the cover source-mismatch. *CoRR* abs/1511.04855 (2015).

[41] Lionel Pibre, Jérôme Pasquet, Dino Ienco, and Marc Chaumont. 2016. Deep learning is a good steganalysis tool when embedding key is reused for different images, even if there is a cover sourcemismatch. In *Media Watermarking, Security, and Forensics*. 1–11.

[42] Kazem Qazanfari and Reza Safabakhsh. 2017. An Improvement on LSB Matching and LSB Matching Revisited Steganography Methods. *CoRR* abs/1709.06727 (2017).

[43] Yinlong Qian, Jing Dong, Wei Wang, and Tieniu Tan. 2015. Deep learning for steganalysis via convolutional neural networks. In *Media Watermarking, Security, and Forensics*.

[44] GMK Ramani, EV Prasad, S Varadarajan, Tirupati SVUCE, and Kakinada JNTUCE. 2007. Steganography using BPCS to the integer wavelet transformed image. *IJCSNS* 7, 7 (2007), 293–302.

[45] Mennatallah M Sadek, Amal S Khalifa, and Mostafa GM Mostafa. 2015. Video steganography: a comprehensive review. *Multimedia tools and applications* 74, 17 (2015), 7063–7094.

[46] H. R. Sheikh and A. C. Bovik. 2006. Image information and visual quality. *IEEE Transactions on Image Processing* 15, 2 (2006), 430–444.

[47] Jeremiah Spaulding, Hideki Noda, Mahdad N Shirazi, and Eiji Kawaguchi. 2002. BPCS steganography using EZW lossy compressed images. *Pattern Recognition Letters* 23, 13 (2002), 1579–1587.

[48] Gandharba Swain. 2018. Digital Image Steganography Using Eight-Directional PVD against RS Analysis and PDH Analysis. *Adv. in MM* 2018 (2018), 4847098:1–4847098:13.

[49] Abdelfatah A Tamimi, Ayman M Abdalla, and Omaima Al-Allaf. 2013. Hiding an image inside another image using variable-rate steganography. *IJACSA* 4, 10 (2013).

[50] Shunquan Tan and Bin Li. 2014. Stacked convolutional auto-encoders for steganalysis of digital images. In *APSIPA*.

[51] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2014. C3D: Generic Features for Video Analysis. *CoRR* abs/1412.0767 (2014).

[52] Guanshuo Xu, Han-Zhou Wu, and Yun-Qing Shi. 2016. Structural design of convolutional neural networks for steganalysis. *IEEE Signal Processing Letters* 23, 5 (2016), 708–712.

[53] Shun Zhang, Liang Yang, Xihao Xu, and Tiegang Gao. 2018. Secure Steganography in JPEG Images Based on Histogram Modification and Hyper Chaotic System. *IJDCF* 10, 1 (2018), 40–53.

[54] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. 2018. HiDDeN: Hiding Data With Deep Networks. *arXiv preprint arXiv:1807.09937* (2018).