VStegNET: Video Steganography Network using Spatio-Temporal features and Micro-Bottleneck

Aayush Mishra*[1]
aayushmishra777@gmail.com
Suraj Kumar*[2]
csksuraj17@gmail.com
Aditya Nigam[1]
aditya@iitmandi.ac.in
Saiful Islam[2]
saifulislam@zhcet.ac.in

[1] Indian Institute of Technology Mandi, India
[2] Aligarh Muslim University Aligarh, India

Abstract

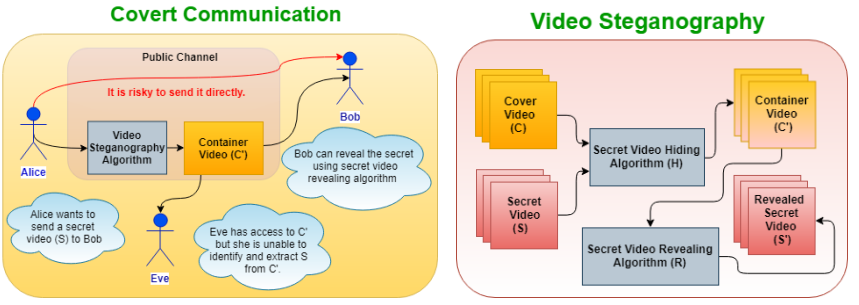
Steganography is the practice of hiding a secret message in a cover message such that the cover stays indiscernible after hiding and only the intended recipients can extract the secret from it. Traditional image steganography techniques hide the secret image into high-frequency regions of the cover images. These techniques typically result in lower embedding ratios and easy detection. In this paper, we propose VStegNET, a video steganography network that extracts spatio-temporal features using 3D-CNN and micro-bottleneck (Hourglass) which is the first of its kind in the literature of video steganography. The proposed network hides $M \times N$ (RGB) secret video frames into same sized cover video frames. We have trained our model on *UCF 101 action recognition* video dataset and evaluated its performance using various quantitative metrics (APD, PSNR, and SSIM) and compared it with previous the state-of-the-art. Furthermore, we have also presented a detailed analysis, supporting the proposal's superiority over image steganography models. Finally, several standard steganalysis tools like StegExpose, SRNET, etc. have been used to justify the steganographic capabilities of VStegNET.

1 Introduction

Data on the Internet is growing exponentially, and it is vital to ensure the security and ownership of that data. Steganography and Cryptography [12] are practices of ensuring the security of confidential data on public channels. Unlike cryptography, steganography provides covert communication which does not attract intruders. Steganography is the practice of concealing a secret message into an ordinary cover message such that only the intended recipients are aware of the secret message [1, 9, 10, 23, 33].

The basic principle and pipeline of video steganography is shown in Fig.1. Traditional algorithms in steganography mainly focus on images and use distortion functions to hide the

*Both authors contributed equally.



(a) (b)

Figure 1: Covert communication and Basic Video Steganography pipeline.

fixed payload into images. One of the earliest and most simple methods is the Least Significant Bit (LSB) replacement steganography [19] where secrets are hidden by replacing LSBs of covers. However, this is prone to statistical analysis as one can easily find the secret by analyzing the LSBs of covers. Later, the research shifted towards designing better distortion functions for steganography. The highly undetectable steganography (HUGO) [23] uses a distortion function based on the weighted difference of Subtractive Pixel Adjacency Matrix (SPAM) [22] feature vectors. Wavelet Obtained Weights (WOW) [9] uses an additive distortion function to hide data in highly textured or noisy regions of the cover. S-Uniward [10] uses universal wavelet relative distortion to embed the secret in an arbitrary domain, But these conventional methods are generally not able to hide more than 0.4 bpp payload into the images and are also unable to handle adversarial attacks.

Recent approaches involve convolutional neural networks [1, 30, 31] which are capable of hiding a significant number of bits-per-pixel (bpp) in cover images without any abrupt change in its appearance. Zhu *et al*. [33] proposed a CNN based method to hide fixed bit messages into greyscale images. There are some generative adversarial network (GAN) [6] based methods [5, 7, 13, 29, 32] that use a discriminator network to en-corporate an adversarial loss in training to improve quality as well as capability to handle adversaries.

Generally, videos would be capable of embedding more data in comparison to images, as they have a temporal component which typically comprises of small optical motion making the data redundant and easier to store. Also, thanks to the fast network technologies today, video is the most popular form of digital media consumed on the Internet. Therefore, it is a more suitable medium to embed secret messages. Weng *et al*. [30] introduced the concept of convolutional neural network (CNN) to video steganography for the first time. They did it with temporal residual modeling for utilizing properties of videos but used a 2D-CNN, which may not be as suitable as 3D-CNNs due to lack of motion modeling [28]. As 3D convolution has been used for medical image segmentation [14], human action recognition [15] and video classification [16] and therefore, it can be used in steganography as well.

In this paper, we have exploited 3D-CNN to learn spatio-temporal feature for video steganography. The key contributions are as follows:

- Proposed a novel framework for video steganography using 3D-CNN and micro-bottleneck (VStegNET).
- Rigorously tested and analyzed VStegNET's ability to handle adversarial attacks by conventional as well as deep learning-based steganalysis tools.
- Quantitative performance evaluation using standard performance metrics at the frame,

video, and dataset level to compare with recent state-of-the-art NIPS-Image model [1] and Video-2018 model [30].

- Experimental analysis of VStegNET via. capacity, failure-cases, drawbacks, and layer-wise activation visualizations to validate VStegNET performance and generalization.

The rest of the paper is organized as follows: Section 2, presents the proposed VStegNET. The results, performance, and comparison with previous methods have been discussed in Section 3. Finally, Section 4 concludes our proposal.

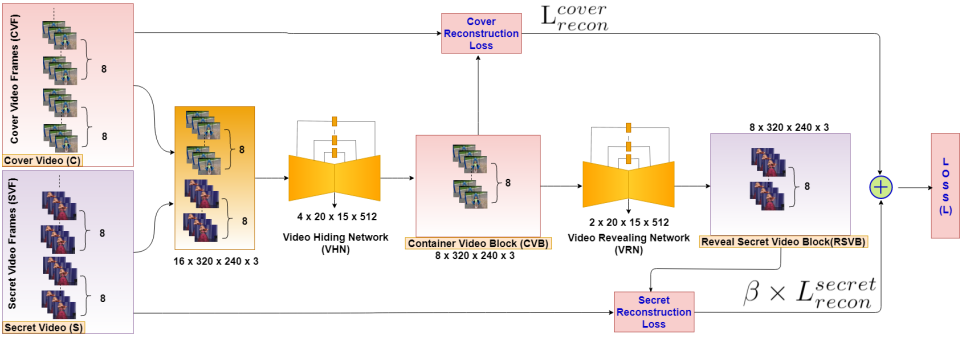


Figure 2: VStegNET proposed architecture: *VHN* receives 8 *CVF* and 8 *SVF* as input, and produces a *CVB* of 8 frames as output. *VRN* reconstructs a *RSVB* of 8 frames from *CVB*. *VHN* and *VRN* are jointly trained to minimize loss L.

2 The Proposed Video Steganography VStegNET Model

Previously, image models have used CNNs to hide data in images [1]. They can also be used to hide data in any video frame-by-frame. Due to a spatio-temporal relation between video frames, we propose VStegNET comprising of a 3D-CNN based autoencoder network for data hiding as well as revealing for the task of steganography. In this section, we describe the proposed VStegNET, as shown in Fig. 2.

[2.1] Network specifications: Our model consists of two connected networks whose specifications are described in Fig. 3. The first one is the Video Hiding Network (*VHN*), which is a 3D-CNN based on Hourglass [20] network. It extracts spatio-temporal features from temporally concatenated cover and secret frames (*CVF* and *SVF*) to construct a block of container video frames (*CVB*) as output. The second is the Video Revealing Network (*VRN*) which is similar to *VHN* in architecture but takes *CVB* as input and produces a block of revealed secret *RSVB* frames as output.

[2.2] Network design: As seen in Fig. 3, the input dimensions of *VHN* and *VRN* are $16 \times 320 \times 240 \times 3$ and $8 \times 320 \times 240 \times 3$ respectively. Since 3D-CNN takes multiple but fixed numbers of frames as a clip, we decided on 8 frames each from cover and secret video to generate a training sample, empirically. A larger number of frames increases the amount of new information present in a training sample. This makes training the network difficult or requires a bigger network. A smaller number of frames does not allow the network to exploit its potential of cramming more data efficiently in frames having redundant information. The networks *VHN* and *VRN* are 3D-CNN based autoencoders utilizing micro-level

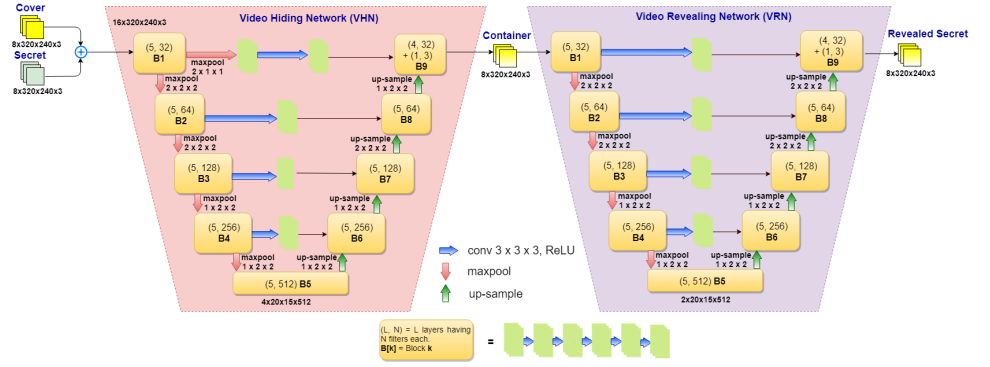


Figure 3: Detailed VStegNET's Hiding and Revealing Network Architecture

bottle-neck features [20] designed to capture feature information at each scale, in local as well as global context. The VHN takes temporally concatenated cover and secret frames as input to maintain temporal sequences for 3D-CNN as well as to spread the secret in all cover frames, rather than hiding frame-by-frame. Skip connections are used to preserve the spatio-temporal information at each resolution. These features are provided through another convolutional layer to the up-sampling layers for facilitating reconstruction optimally. This makes it different from a simple U-Net architecture [24]. The bottleneck size of VHN is $4 \times 20 \times 15 \times 512$ while that of VRN is $2 \times 20 \times 15 \times 512$ adjusted empirically over the training data. Each 3D-convolutional block extracts spatio-temporal features and passes them to the next block after max-pooling to a lower resolution, both spatially and temporally. Every convolution kernel is of size $3 \times 3 \times 3$ with ReLU [18] activation function, and feature size only changes after max-pooling and up-sampling layers. After reaching the lowest resolution, the network has a sequence of up-sampling layers with skip connections where features are concatenated across the channel. The network topology is symmetric, for smooth and fast autoencoder learning. To reach the output resolution, one convolution layer of $3 \times 3 \times 3$ with 3 kernels are applied at the last layer to reconstruct the desired output frames. The only difference between the architecture of VHN and VRN is one extra-temporal pooling, in order to make it suitable for corresponding concatenation.

[2.3] Significance of 3D-CNN and Hourglass Network: The spatio-temporal features can relate time as well as space together. In addition to the local spatial features present in a frame, it also correlates adjacent frame features and can represent the video bottleneck much more effectively. However, 2D-CNNs, do not capture such temporal dependencies between adjacent frames. Another major problem of deep networks is over-fitting and vanishing gradients [27]. The micro-bottleneck based Hourglass networks have long as well as short skip connections that are capable of addressing these issues by easing the flow of gradients to distant layers [20]. Lastly, as auto-encoders shrink inputs to very low resolutions, skip connections help in reconstruction.

[2.4] Loss Function: The loss function of VStegNET is defined in Eq.1.

$$Loss(C, C', S, S') = L_{recon}^{cover} + \beta * L_{recon}^{secret} = \| C - C' \|_2 + \beta * \| S - S' \|_2 \quad (1)$$

where C, C', S, S' are cover, container, secret and revealed secret respectively. The parameter β is introduced to bias the reconstruction of container frames, as it is the primary objective of any covert communication. This regularizes VStegNET's learning.

3 Experimental Analysis

[3.1] Dataset specifications: The UCF101 dataset [25], contains 13,320 videos with plenty of variations in action, camera motion, background cluttering, object appearance and pose, object scale, illumination etc. Each video frame is of the shape $(320 \times 240 \times 3)$.
[3.2] Video Sampling: Videos are split into frames, and then one video each for cover and secret are selected from the training set, at random. As every video consists of the different number of frames, temporal equalization has been done as follows. Assuming N_1 and N_2 are number of frames in the randomly chosen cover and secret video, the highest multiple of 8, that is lower than the minimum of N_1 and N_2 is choose as N, defined as $N = min(N_1, N_2) - min(N_1, N_2)\ mod\ 8$. N cover and secret frames are then fed to VStegNET.
[3.3] Training and Testing specifications: End-to-end training has been done on NVIDIA Geforce 1080 Ti using Adam [17] optimizer with $\alpha = 10^{-4}$, to minimize the reconstruction loss (Eq.1). Out of all dataset videos, 10,000 are used to generate training cover, and secret pairs (total possible pairs = $C_2^{10,000}$) and remaining are used for testing. It was observed that VStegNET converged rapidly after training on just 5000 video pairs and generalized well over unseen test pairs. While testing, 500 video pairs were sampled on random, multiple times from the test-set with 140 frames per video on average. The results reported in order to justify VStegNET's accuracy and generalization, are averaged over all these test samples.
[3.4] Results: VStegNET was trained for two values of β viz. 0.75 and 1.0. Few results (good, median, and poor) of our model with $\beta = 0.75$ on the test set are shown in Fig. 4. All other comparisons that follow are also made with that model unless stated otherwise. Few bad results from our model are analysed and explained in Fig. 10.

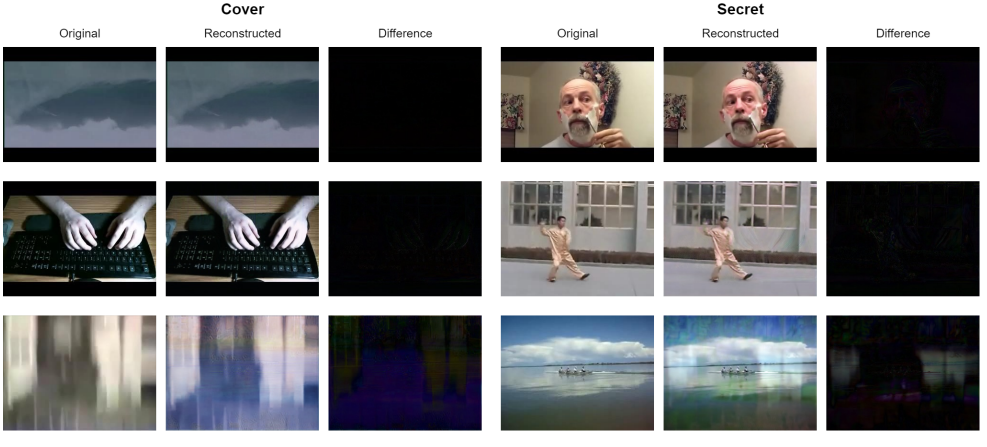


Figure 4: First row: A good test case having no visual artifacts in the reconstructed cover and secret. Middle Row: A medium case having almost no visual artifacts. Last row: A bad case when there are abrupt changes in frames. For details, see Fig. 10.

[3.5] Quantitative Performance Analysis: The performance evaluation using three metrics viz. Average Pixel Discrepancy (APD) (average absolute pixel difference (out of 255) between original and reconstructed images, per pixel), Peak Signal to Noise Ratio (PSNR) [11] and Structural Similarity Index Measure (SSIM) [11], corresponding to Fig. 4 can be seen in Table. 1, for frame as well as video level. The overall mean results for all frames from all test sets are also summarized in the Table. 1. The (μ, σ) of APD, PSNR and SSIM for (C, C')

and (S, S') are [(3.17, 1.25), (5.77, 2.00)], [(35.67, 3.20), (30.07, 2.66)], [(0.94, 0.04), (0.92, 0.04)] respectively. The trade-off between secret and cover reconstruction performance can be made by adjusting β. Fig. 5 shows the histograms of these performance metrics obtained on the overall aggregated test set. A single frame anomaly is not representative of the overall video quality, which is usually good. Frames which have high PSNR and SSIM and low APD values show less visual artifacts, in general.

[3.6] Comparative Performance Analysis: In Table 2 , we report the comparative results of traditional LSB [19], along with with an image-level deep learning model recently proposed at NIPS [1] and a video level 2018 model [30] with VStegNET. APD was the only metric on which we could compare all these models. A sample visual comparison of our model's outputs of VHN and VRN with [1] has been shown in Fig. 9. Note that all results of the image model are calculated after training the implementation of [1] on our dataset until convergence.

Table 1: The APD, PSNR and SSIM values of the test samples shown in Fig. 4.

	APD				PSNR				SSIM			
	(C, C')		(S, S')		(C, C')		(S, S')		(C, C')		(S, S')	
	Frame	Video	Frame	Video	Frame	Video	Frame	Video	Frame	Video	Frame	Video
Good	0.96	**1.74**	3.80	**4.56**	44.67	**40.74**	33.38	**31.78**	0.98	**0.97**	0.98	**0.97**
Medium	3.07	**2.69**	4.39	**4.96**	36.03	**36.72**	32.97	**31.66**	0.88	**0.90**	0.95	**0.94**
Bad	19.30	**3.57**	13.28	**7.16**	20.04	**35.01**	21.78	**28.84**	0.87	**0.96**	0.87	**0.92**
Testset Mean	**3.17**		**5.77**		**35.67**		**30.07**		**0.94**		**0.92**	

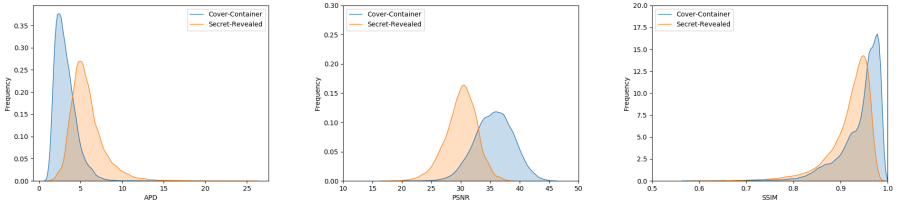


Figure 5: Histograms showing the APD, PSNR and SSIM distribution on test set.

[3.7] Where is the Secret encoded? Our model stores the secret frames not only spatially but also temporally within container frames. When container produced by VStegNET are fed into the revealing network of the image steganography model [1] (trained on our dataset until convergence), it cannot decode the embedded secret from containers as shown in Fig. 8. This, however, also means that if someone mishandles (crops, adds noise, deletes frames, etc.) our container video, one can lose the hidden secret. But VStegNET can be naturally extended to handle these attacks using standard procedures of redundancy spatially and temporally. For analysis of spatial and temporal storage of secret frames in the cover frames refer Fig. 6.

[3.8] What does the model see? Fig. 7 shows the journey of adjacent cover and secret frames through our model's convolutional layers (i.e., activation map). We can see the gradual and uniform mixing/blending of cover and secret frames. These activation maps provide meaningful insights into how our model perceives its input and what changes does it make at each stage.

[3.9] Capacity: The embedding capacity in steganography is usually defined in term of bits per pixel (bpp) which is the number of bits required to store one pixel of an image.

Figure 6: Non-sequential randomly selected frames (first row) fed to VStegNET, produce poor container and revealed secret frames (second row) showing that secret is hidden in multiple frames instead of just one using a 3D-CNN, showing resistance to secret revelation using 2D-CNN based models. Image model [1] does not face this problem (third row), but is prone to secret revelation.

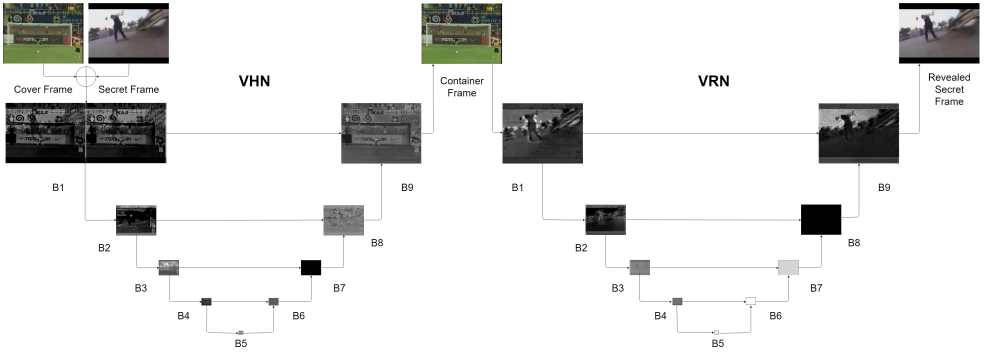


Figure 7: Activation maps produced by convolution filters from the last layers of each block.

Figure 8: Revealing network of [1] produces useless results when containers generated by VStegNET are fed to it. This demonstrates that secrets hidden by VStegNET cannot be revealed by other image-based networks.

In conventional steganographic methods, a fixed number of bits of the secret message were hidden in images, which was typically less than 0.4 bpp [5, 9, 10, 23]. This is a small

amount of information, undermining the ability of images to store large amounts of data. The average bpp of secret test data was **10.51**, that got hidden in the cover. Currently, we do not address mishandling of containers e.g., noising, cropping, which requires redundant spatial and temporal embedding. That would obviously reduce the embedding capacity.

Table 2: APD values: The values for LSB and video model are taken from [30] as their code was not available. The image model [1] was trained and tested on our dataset.

Model	$\|\| C - C' \|\|$	$\|\| S - S' \|\|$
LSB [19]	6.64	8.64
NIPS, 2017 [1]	6.31	4.97
Video, 2018 [30]	3.80	5.84
Ours ($\beta = 0.75$)	**3.17**	5.77
Ours ($\beta = 1.0$)	3.56	**4.89**

Table 3: Classification results: SVM was trained with default parameters. Pre-trained ResNet50 and Inception-v3 models were fine-tuned for the task.

Classifier	Accuracy
SVM on SPAM	53.3
ResNet50	49.7
Inception-v3	55.4

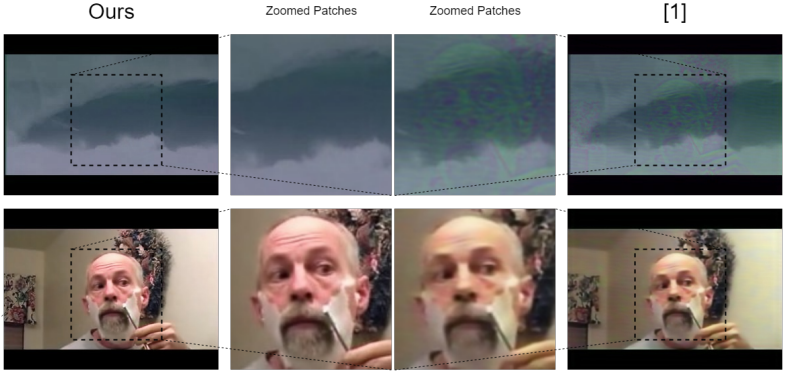


Figure 9: Visual comparison of our model with [1]. First row: Containers, Second Row: Revealed Secrets. A 135×135 patch is zoomed to show superior results.

[3.10] Steganalysis: The test of robustness against discoverability is a parallel field of steganography called steganalysis. We used the well known steganographic feature extractor tool called Subtractive Pixel Adjacency Matrix (SPAM) [21] to extract features from our container and cover frames. It produces a 686-dimensional vector for each input image. These feature vectors can then be trained on a classifier to identify traces of steganography. We obtained SPAM features of 20,000 cover and container frames from the test-set and tried training a Support Vector Machine (SVM) [4] to distinguish between these feature vectors. We also tried using the well-known ResNet50 [8] and Inception-v3 [26] models pre-trained on ImageNet dataset. We used the deep conv features from these trained networks and tried training a neural network to distinguish between those features. Both these binary classification tasks were aimed to see if existing techniques can be used to find features from the images generated using VStegNet that can differentiate them from normal images. The results of these experiments are summarized in Table 3, which show nothing better than random guessing. This shows our model's robustness against steganalysis. This is for the case

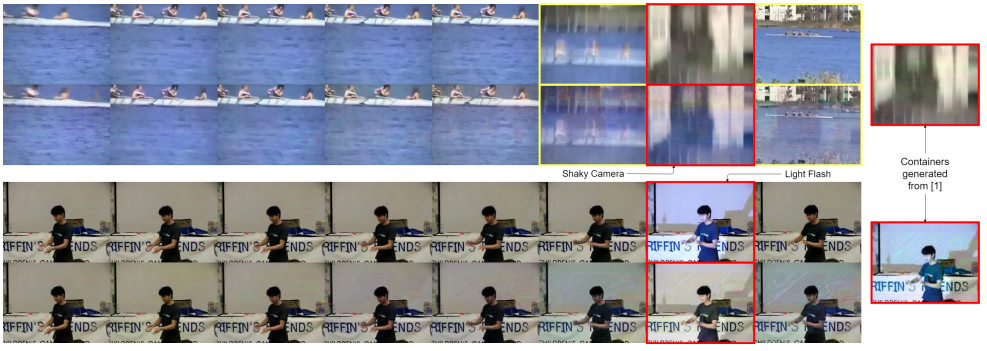
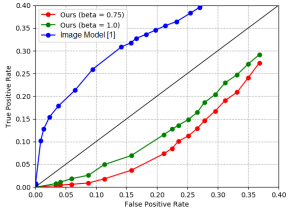


Figure 10: Highlighted (Yellow), Cover frames (First and Third row) are significantly different from adjacent frames, due to a **shaky camera** and **light flash**. The 3D conv-filter sees three frames at a time, utilizing optical flow in the frames. Therefore the generated containers (Second and Fourth row) get severely affected. The last column shows containers (red) generated using [1] which are better than corresponding frames from VStegNET because of no temporal information flow in 2D-CNNs.

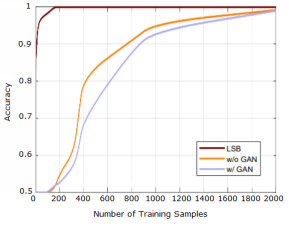
when labeled container frames generated from VStegNET, are not available, and someone tries to inspect an image for steganography using some outlier detection techniques.

However, in case if labeled data is available (from VStegNET), it is possible for deep CNNs to act as adversaries and differentiate between cover and container frames. For this case too, we trained ResNet50 and Inception-v3 from end to end. We also trained a new deep residual network for steganalysis, SRNet [3] for this task. We plotted the number of training samples required for these adversarial networks to get an almost perfect accuracy of detection. This is done in comparison with [30] (see Fig. 11(b),(c)). Here also, the proposed VStegNET requires many more images as compared to other state-of-the-art techniques.

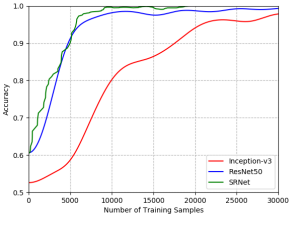
Finally, we also used a popular steganalysis tool called StegExpose [2] that seeks out the information hidden in the LSBs of any image. We varied the threshold used by this algorithm for the detection of steganographic images and plotted the ROC curve, for a fair comparison of our results with [1] (NIPS image model) as shown in Fig. 11(a). One can observe that again, VStegNET performs significantly better than [1]. This exhibits that our model is not restrictive to hide data in LSBs of cover images.



(a) Comparative ROC curves for StegExpose of VStegNET and [1]



(b) ∼2000 training samples required for an adversary to differentiate container and cover [30]



(c) VStegNET requires much more samples to train a perfect adversary.

Figure 11: Comparative ROC curves and robustness analysis against adversarial attack.

4 Conclusion and Future Work

To the best of our knowledge, VStegNET is the first of its kind in the literature of video steganography. We have shown and compared the results of 2D-CNN based models with VStegNET. We have demonstrated our model's qualitative and quantitative performance and also its anti-steganalysis abilities using various standard techniques. There can be several natural extensions/improvements possible to our model such as embedding more data in the covers, by modeling the redundancy in secret data separately, embedding other media types like images, text, audio, etc. and adding adversarial loss for more resistance to steganalysis. An advanced improvement would be to tackle container mishandling like introducing noise, compressing or cropping video spatially and temporally, etc. by repetitive storage of secret in the cover, both spatially and temporally, on which we plan to work ahead.
We have also made our code and other supplementary reading material for a detailed experimental analysis, public on Github. A short video showcasing our model's performance can be seen on YouTube.

References

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

[2] Benedikt Boehm. Stegexpose - A tool for detecting LSB steganography. *CoRR*, abs/1410.6656, 2014.

[3] M. Boroumand, M. Chen, and J. Fridrich. Deep residual network for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 14(5):1181–1193, May 2019. ISSN 1556-6013. doi: 10.1109/TIFS.2018.2871749.

[4] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Mach. Learn.*, 20(3): 273–297, September 1995. ISSN 0885-6125. doi: 10.1023/A:1022627411411.

[5] Shiqi Dong, Ru Zhang, and Jianyi Liu. Invisible steganography via generative adversarial network. *CoRR*, abs/1807.08571, 2018.

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

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

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[9] Vojtech Holub and Jessica J. Fridrich. Designing steganographic distortion using directional filters. *2012 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 234–239, 2012.

[10] Vojtěch Holub, Jessica Fridrich, and Tomáš Denemark. Universal distortion function for steganography in an arbitrary domain. *EURASIP Journal on Information Security*, 2014(1):1, Jan 2014. ISSN 1687-417X. doi: 10.1186/1687-417X-2014-1.

[11] Alain Horé and Djemel Ziou. Image quality metrics: Psnr vs. ssim. *2010 20th International Conference on Pattern Recognition*, pages 2366–2369, 2010.

[12] Zenon Hrytskiv, Sviatoslav Voloshynovskiy, and Yuriy B. Rytsar. Cryptography and steganography of video information in modern communications. 1998.

[13] Donghui Hu, Liang Wang, Wenjie Jiang, Shuli Zheng, and Bin Li. A novel image steganography method via deep convolutional generative adversarial networks. *IEEE Access*, 6:38303–38314, 2018.

[14] V. Jain, B. Bollmann, M. Richardson, D. R. Berger, M. N. Helmstaedter, K. L. Briggman, W. Denk, J. B. Bowden, J. M. Mendenhall, W. C. Abraham, K. M. Harris, N. Kasthuri, K. J. Hayworth, R. Schalek, J. C. Tapia, J. W. Lichtman, and H. S. Seung. Boundary learning by optimization with topological constraints. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2488–2495, June 2010. doi: 10.1109/CVPR.2010.5539950.

[15] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1): 221–231, Jan 2013. ISSN 0162-8828. doi: 10.1109/TPAMI.2012.59.

[16] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1725–1732, June 2014. doi: 10.1109/CVPR.2014.223.

[17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[18] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 807–814, USA, 2010. Omnipress. ISBN 978-1-60558-907-7.

[19] D. Neeta, K. Snehal, and D. Jacobs. Implementation of lsb steganography and its evaluation for various bits. In *2006 1st International Conference on Digital Information Management*, pages 173–178, Dec 2007. doi: 10.1109/ICDIM.2007.369349.

[20] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.

[21] T. Pevny, P. Bas, and J. Fridrich. Steganalysis by subtractive pixel adjacency matrix. *IEEE Transactions on Information Forensics and Security*, 5(2):215–224, June 2010. ISSN 1556-6013. doi: 10.1109/TIFS.2010.2045842.

[22] Tomáš Pevny, Patrick Bas, and Jessica Fridrich. Steganalysis by subtractive pixel adjacency matrix. *IEEE Transactions on information Forensics and Security*, 5(2):215–224, 2010.

[23] Tomáš Pevný, Tomáš Filler, and Patrick Bas. Using high-dimensional image models to perform highly undetectable steganography. In *Proceedings of the 12th International Conference on Information Hiding*, IH'10, pages 161–177, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-16434-X, 978-3-642-16434-7.

[24] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[25] Khurram Soomro, Amir Roshan Zamir, Mubarak Shah, Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, page 2012.

[26] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.

[27] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. C3D: generic features for video analysis. *CoRR*, abs/1412.0767, 2014.

[28] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 4489–4497, Washington, DC, USA, 2015. IEEE Computer Society. ISBN 978-1-4673-8391-2. doi: 10.1109/ICCV.2015.510.

[29] Denis Volkhonskiy, Ivan Nazarov, Boris Borisenko, and Evgeniy Burnaev. Steganographic generative adversarial networks. *CoRR*, abs/1703.05502, 2017.

[30] Xinyu Weng, Yongzhi Li, Lu Chi, and Yadong Mu. Convolutional video steganography with temporal residual modeling. *CoRR*, abs/1806.02941, 2018.

[31] Pin Wu, Yang Yang, and Xiaoqiang Li. Stegnet: Mega image steganography capacity with deep convolutional network. *CoRR*, abs/1806.06357, 2018.

[32] Kevin Alex Zhang, Alfredo Cuesta-Infante, Lei Xu, and Kalyan Veeramachaneni. Steganogan: High capacity image steganography with gans. *CoRR*, abs/1901.03892, 2019.

[33] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XV*, pages 682–697, 2018. doi: 10.1007/978-3-030-01267-0_40.