# Exploring Advanced Deep Learning Methods for Image Steganography: New Approaches to Concealing Videos Within Videos

Omar Esteban Vargas Salamanca[1], Kevin Cohen Solano[2], and Julian Camilo Mora[3]

[1] Universidad de los Andes, Bogotá, Colombia
o.vargas@uniandes.edu.co
[2] k.cohen@uniandes.edu.co
[3] j.morav@uniandes.edu.co

**Abstract.** Steganography is the art of concealing information within another medium, traditionally involving the embedding of text within images. However, an increasingly relevant application is the concealment of one image within another, known as image-in-image steganography. Extending this concept further, this research introduces the innovative approach of video-in-video steganography, which applies image-in-image techniques to conceal entire video sequences within other videos. By leveraging the capabilities of Deep Learning, specifically convolutional neural networks, improving the undetectability and efficiency of our steganographic methods. Focus on refining how visual data is concealed within other multimedia content leads to more structured and cohesive embedding strategies. The models were trained using instance the popular ImageNet dataset [1], showcasing proficiency in embedding secret images and video sequences within host media, undetectably. This work evaluated the performance of our models using Structural Similarity Index Measure (SSIM) to assess their capacity to encode and decode secret data effectively. The primary contribution of our study is the development of a system capable of performing steganography between two videos of the same length, thereby offering a novel perspective within the field of traditional steganography. This exploration not only pushes the boundaries of conventional techniques but also paves new pathways for secure visual data transmission, highlighting the transformative potential of convolutional neural networks in revolutionizing steganography.

**Keywords:** Steganography, Convolutional Neural Networks, Deep Learning, Encoding, Decoding.

## 1 Introduction

Steganography, the art of hiding messages within a medium to ensure they remain unnoticed, involves complex layers of analysis and processing [2]. Traditionally, a hidden message is embedded within a carrier—often an image—forming what is known as a stego medium or container, described by the equation:

$$stegomedium = hiddenmessage + carrier \tag{1}$$

This process, though simple in concept, has evolved significantly due to technological advancements, expanding the range of potential carriers—often referred to as covers in other research—to include digital media such as images, audio, and video [3] [4]. Innovations have particularly enhanced the method of image-into-image steganography, where one image is concealed within another, ensuring both undetectability and reconstructibility [5].

Inspired by these advancements and the pioneering work by Wani, M. A and Sultan, B. in *Deep learning based image steganography: A review. WIREs Data Mining and Knowledge Discovery* [6], this research introduces a novel approach: video-into-video steganography. This method utilizes Convolutional Neural Networks (CNNs) to encrypt frames of two videos of identical length, applying and extending principles from image-in-image techniques to video sequences. By determining optimal placements for the secret information and efficiently encoding and decoding it, CNNs enhance the utility of traditional methods for dynamic media contexts. Leveraged the diverse image sizes and qualities available in an instance of the popular dataset ImageNet [1] to develop and train models that proficiently handle high-quality video content with minimal perceptual distortion. This approach not only challenges the traditional boundaries of steganography but also significantly expands upon the foundational concepts introduced in the use of Machine Learning in Steganography. It specifically applies these principles to color videos, which are effectively sequences of images. For a detailed illustration of how these components interact within our system, see Figure 1.

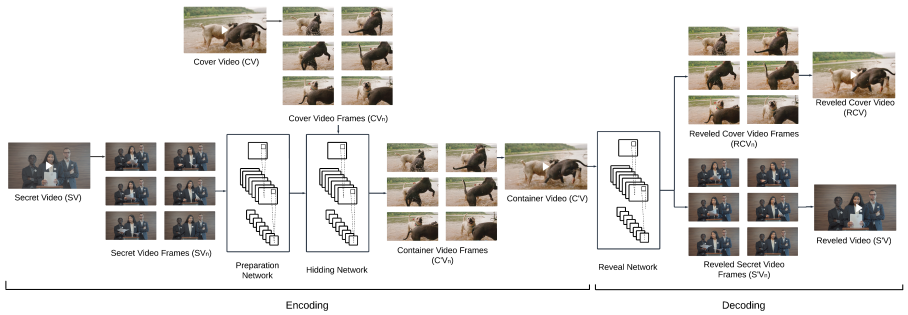


**Fig. 1.** Three components of the full CNN system.

To evaluate the effectiveness of our models, this study employs Structural Similarity Index Measure (SSIM), which is crucial for assessing the integrity and quality of stego videos [5]. Our results demonstrate that our CNN-based models not only achieve high rates of data embedding but also maintain the fidelity of the cover video, significantly improving upon traditional steganographic techniques.

This study's contributions are significant, introducing a robust framework for video-in-video steganography that utilizes CNNs to seamlessly embed large volumes of data within videos. By exploring the trade-offs between embedding capacity and visual quality, our work offers viable solutions for real-world applications where security and quality are paramount, thereby expanding the boundaries of traditional steganography and setting the stage for future secure digital communication research.

## Related Work

This section reviews the existing literature on steganographic techniques, with a particular focus on the evolution from traditional methods to advanced applications involving deep learning techniques and multimedia data.

First of all, it is pertinent to point out that the main studies involving the relationship between steganography and deep learning are based on classifiers. For example, the most known steganography techniques such as LSB of images are used and deep learning models and traditional ML techniques are used to detect if given an image contains any message or type of hidden content. On the other hand, the objective of this research is to find and explore models based on deep learning that are alternatives to the most common steganography techniques in which integrity is not the priority [2].

On the other hand, the use of deep learning models for image steganography are mainly based on GANs and in general show acceptable results in terms of metrics such as Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index (SSIM), one of the limitations they may have refers to amount of computational resources and data and training time required to get a good model. On the other hand, the architecture proposed in this research approaches to propose a simpler encoder-decoder based convolutional network architecture that minimizes both training and prediction time of the image with steganography [7].

Returning to contemporary times, the digital revolution has significantly strengthened the field of steganography, particularly when combined with deep learning techniques. This combination aims to encrypt messages within containers more efficiently and undetectably. For instance, in a recent study, Hashemi (2022) [8] explains how to create image steganography utilizing deep learning, specifically convolutional autoencoders combined with ResNet architecture. This method addresses the limitations of traditional steganography techniques, such as low capacity and robustness, by employing a reverse ResNet architecture to extract the hidden image from the stego image. Similarly, Hayes, J., & Danezis, G. (2017) [9] explore ResNet-based architectures to extract stego images, reporting an SSIM of 0.98, indicating that the obtained container images are nearly identical to the cover image without encrypted secrets, which is a significant indicator of the success rate of these models. This sets a goal for us to achieve and even surpass in our research. It is worth mentioning that advances in steganography using machine learning have been developing for years, as can be seen in the

case of HUGO, a highly popular method in the field of steganography. HUGO is specialized in embedding for spatial-domain digital images and was proposed by Pevný, T., Filler, T., & Bas, P. (2010) [10].

With the rise of deep learning, researchers have explored the use of convolutional neural networks to create innovative ways of embedding information in containers with multiple digital formats. Notably, Baluja's work [6] on image-in-image steganography using deep learning set a new benchmark for the field, inspiring further research in the field of encoding digital images inside of other digital images in an undetectable way [11]. Finally, it is noteworthy to highlight recent studies in the state of the art regarding the application of steganography within videos to encode messages in text format, as explained by Kunhoth, J., Subramanian, N., Al-Maadeed, S., & Bouridane, A. (2023) [12]. However, there is limited information regarding the utilization of the computational capabilities mentioned in this section for video-into-video steganography. This lack of information motivates us to generate this knowledge and lead our own research to contribute to this field.

## Architecture of the Model and Application Explanation

In this project, an autoencoder architecture tailored for steganography tasks is proposed. The model consists of three primary components: the Preparation Network, Hiding Network (Encoder), and Reveal Network. The overarching objective is to encode information from a secret image (S) into a cover image (C), thereby generating C', which closely resembles C, while retaining the capacity to decode information from C' to reconstruct the secret image (S').

The Preparation Network is responsible for preprocessing data from the secret image to be merged with the cover image input for the Hiding Network. Subsequently, the Hiding Network transforms this combined input into the encoded cover image (C'). Finally, the Reveal Network decodes the hidden information from C' to generate the decoded secret image (S').

To enhance stability during decoding, noise is introduced prior to the Reveal Network, as referenced in literature. Although specific architectural details for the three networks were not originally delineated by the paper's author, a structure with 5 layers comprising a total of 65 filters, distributed across 3x3, 4x4, and 5x5 filter sizes, is implemented. Notably, the Preparation Network is simplified to consist of only 2 layers following the same filter distribution.

The loss model used in this study was trained over 100 epochs, with a redefined loss function aimed at comparing both the generated $S'$ and $C'$ images, using a value of $\beta = 1.0$.

For the reveal network, the loss function was formulated as $\beta \times |S - S'|$, where $S$ represents the ground truth secret image and $S'$ denotes the predicted secret image.

For the full model, encompassing both preparation and hiding networks, the loss was computed as the sum of two terms: $|C - C'|$ to evaluate the similarity

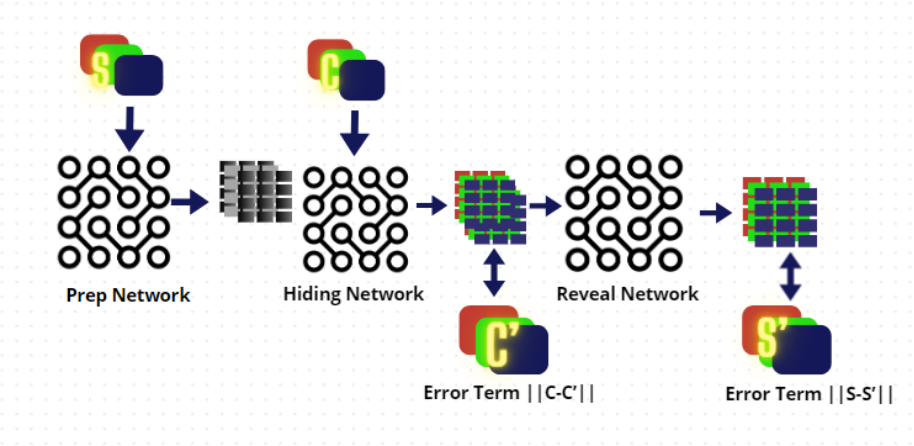


**Fig. 2.** Architecture of Steganography Autoencoder Model.

between the cover images, and $\beta \times |S - S'|$ to assess the fidelity of the secret images.

### Justification for Excluding Component Analysis

While techniques such as PCA or ICA could potentially reduce the complexity of the problem, they were not included in this model. This decision is based on several key considerations:

**Integrity of Reconstructed Image:** Techniques like PCA or ICA can reduce dimensionality, but they may also lead to the loss of crucial information necessary for accurately reconstructing the secret image. In steganography, ensuring a faithful representation of the input data is essential for the accurate recovery of the hidden information. The model prioritizes maintaining the full integrity of the secret image, even if it means not reducing the complexity beforehand.

**Representation in the Input:** The fidelity of the input representation is vital for the effectiveness of the steganographic process. By not applying dimensionality reduction techniques, the Preparation Network ensures that all relevant information from the secret image is retained and utilized in the encoding process. This comprehensive representation helps the Hiding Network encode the secret image into the cover image more effectively.

**Robustness and Visual Quality:** The autoencoder is designed to be robust against variations and noise in the input data, enhancing its stability and performance without the need for prior dimensionality reduction. Additionally, maintaining the visual quality of the encoded cover image (C') is critical to avoid detection. Techniques like PCA or ICA might alter the visual characteristics of the cover image, increasing the risk of detection.

### Usage of the Model for Developing a Video Steganography Application

Once the steganography model was properly serialized and ready for being used freely, the next step was to test it with video input formats. Firstly, as the model only receives as input two images of 124×124 pixels at a time, it was necessary to come up with a way for the app to process two entire videos of any length and resolution. The process of processing the video can be seen in 4.

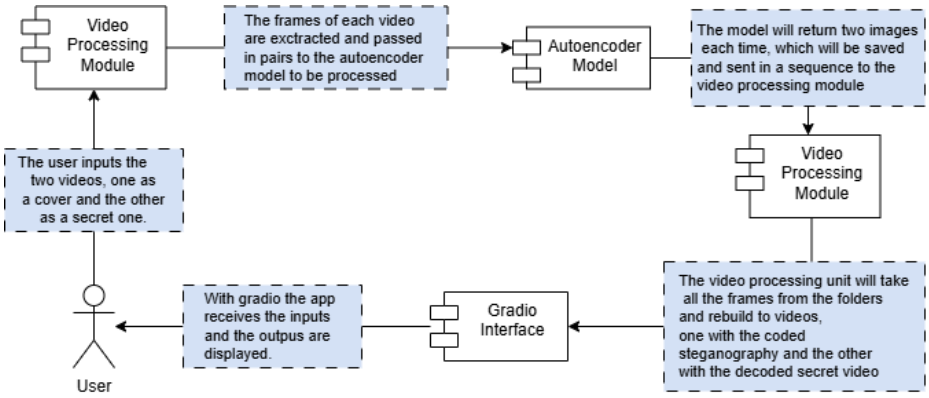


**Fig. 3.** Diagram that shows the process for processing a video.

As a general approach to this process, the app takes each of the videos, and by using the python library "cv2" their frames are extracted one by one and stored under the name "framex.jpg" (where x represents the number of the frame) in two different folders, one for each video. Then, the app takes the original folder (that is the one with the frames of the cover video) and iterates in all of its elements, taking each one of the iterations and pairing them with its respective frame counterpart (that comes from the secret video). Each of these pairs is passed to the autoencoder model (that was previously loaded using TensorFlow) and the new frames for the videos are obtained. The frames are stored independently in different folders in order to re-build a video from them when the process is completed. Lastly, the library "cv2" is used again for re-building a video from the processed frames folders. The videos obtained from this process always come with a resolution of 124×124 pixels, meaning that it reduces significantly the native resolution from the videos. Even though this is not optimal, due to computational limitations it was impossible to train the model with images that match the standard video resolution used (ranges from 1280×720 px to 3840×2160 px).

It is also very important to note that when the results of the processing of each frame are obtained, the Structural Similarity Index Measure (SSIM) value between the original frame (previously rescaled to 124×124 px) and the processed

frame is calculated. This metric is really useful for evaluating steganography, as the SSIM takes into account changes in structural information, luminance and contrast. It is designed to more closely model human visual perception.

The framework used for deploying the application is Gradio. The selection of this technology over others was due to the great documentation that exists from it and the accessibility when trying to deploy the model, even in environments like Colab (from Google).

The application also supports the processing of individual images, taking two as an input and returning the processed images and a small graph that shows the noise between the original images and the processed ones.

## Experimentation and Results

In this section, it describes the training process of the model using a dataset consisting of over 6000 images [1], resized to 124x124 pixels. It is crucial to emphasize that ensuring a diverse dataset is paramount for guaranteeing optimal performance of both the encoder and decoder components of the model. This diversity helps mitigate the risk of overfitting and ensures that the model can generalize well to unseen data.

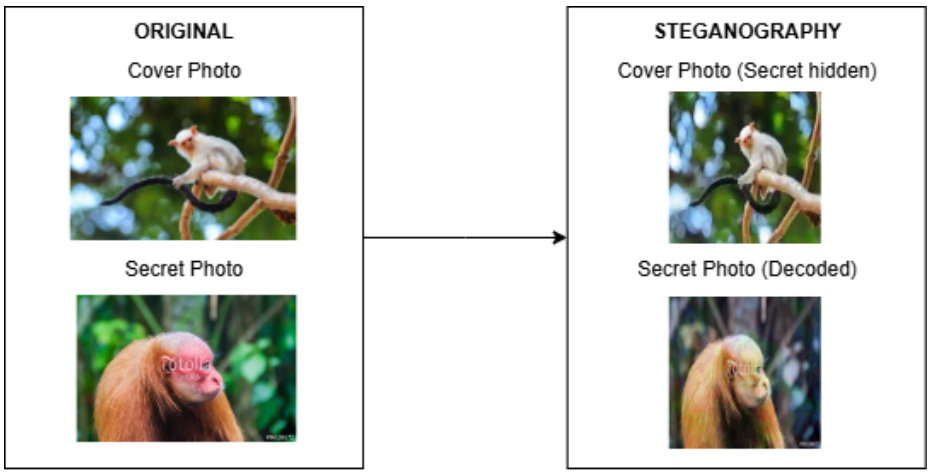


**Fig. 4.** Results of steganography with some example images.

Once satisfactory results are obtained for the decoder component, the model can be effectively utilized with any pair of images, and even videos. Remarkably, to the human eye, the differences between the original and encoded images can be challenging to discern. It is worth noting that in steganography, the model exhibits a remarkable capability to transfer or combine colors in some images, highlighting the CNN's ability to extract relevant features from the images.

The training process involved iteratively optimizing the model's parameters using backpropagation and gradient descent techniques. Data augmentation strategies, such as random rotations, flips, and adjustments to brightness and contrast, were employed to enhance the model's robustness and generalization capabilities. Additionally, early stopping and learning rate scheduling techniques were utilized to prevent overfitting and improve convergence speed.

Overall, the training process aimed to strike a balance between model complexity and generalization ability, ensuring that the model could effectively encode and decode information while maintaining fidelity to the original images. The successful training of the model on a diverse dataset underscores its potential for various applications in image and video processing, with steganography[**?**].

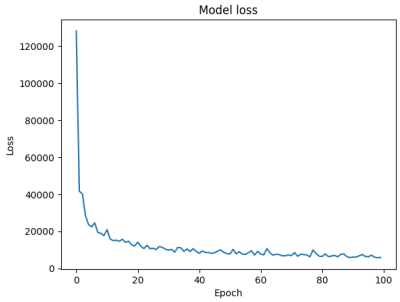


**Fig. 5.** Loss Metrics in a Training Steganography Autoencoder Model

### Challenges in Detecting Encoder-Generated Predictions

One of the main objectives of this study was to assess the difficulty in detecting predictions generated by the encoder, particularly those involved in steganographic processes. Several methodologies were explored to tackle this challenge, each employing distinct machine learning architectures.

Initially, an attempt was made to leverage a network fashioned after the decoder architecture of the autoencoder proposed in this study. The encoder weights were frozen and amalgamated with a dense network housing a sigmoid neuron at the output layer. However, a significant obstacle was encountered when endeavoring to input a solitary image into the model, given the decoder's expectation of dual images. Various attempts were made to address this predicament, including padding the absent tensors with arrays of zeros or ones. Unfortunately, these endeavours yielded suboptimal outcomes.

Subsequent investigations delved into the utilization of pretrained models such as VGG16, MobileNetV2, and a customized variant of VGG16. Despite meticulous training attempts encompassing diverse configurations, satisfactory results remained elusive across all trials. The models exhibited either pronounced overfitting tendencies or achieved performances akin to random guessing (0.5).

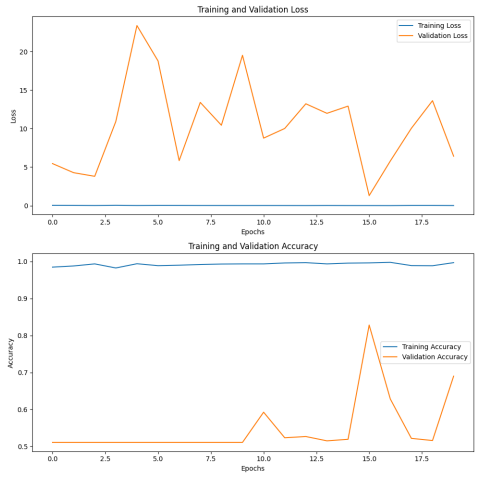


**Fig. 6.** Metrics Binary Classifier Based on MobileNetV2

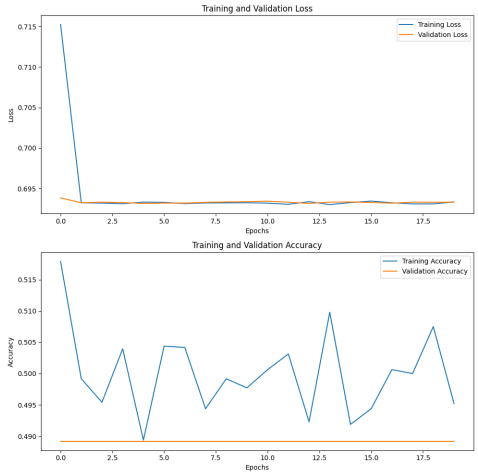


**Fig. 7.** Metrics Binary Classifier Based on VGG16

Confronted with these challenges, an alternative approach was pursued employing a siamese network architecture. This network was engineered to accept both the original image and the image synthesized by the encoder as input. However, formidable hurdles were once again encountered, as models grounded on convolutional or dense networks struggled to attain satisfactory performance levels. Even after exhaustive experimentation involving data augmentation techniques and fine-tuning of hyperparameters like learning rates, the results failed to meet the desired criteria.

Additionally, a convolutional siamese network was also employed, where the inputs were the original image and the one generated by the encoder. Despite these efforts, satisfactory results were not achieved.

The inherent difficulty in detecting predictions generated by the encoder lies in the complex and subtle alterations made to the input image. Unlike traditional generative models, which often introduce noticeable artifacts, the encoder aims to embed information seamlessly within the cover image, making it challenging for classification models to discern between original and encoder-generated predictions. Additionally, the nature of steganographic processes involves encoding information in imperceptible ways, further complicating the task of detection.

### Video Experimentation

Regarding the results with video steganography [**?**], the main metric used was SSIM. Even though things like the noise and absolute pixel difference could be had into account, these metrics wouldn't be different from the ones obtained when originally training and testing the model. The results obtained for both the cover video and the secret video are pretty good, getting mostly values close to one. This means that for the human eye, there isn't much perceivable difference between the original content and the one that has steganography on it. The charts 10 and 11 show the results obtained when testing with two videos of 243 frames.

This graph (10) shows the results for the SSIM in the original cover video and the processed cover video. The range of values doesn't go under 0.98, and it oscillates between hundredths. This result is superb, meaning that the results for the usage of this model are great when hiding videos inside other videos.

For this other graph (11), the results show the SSIM values for the original secret video against the decoded secret video. With the secret video, the results are also interesting, but a little worse than the ones obtained in the cover video. The oscillation of the range goes under 0.86 and up to more than 0.93. This means that the model does a better job hiding the images than recovering them, yet giving an understandable output when recovering the inputs.

The model was trained on 3000 samples and took approximately 1 hour and 20 minutes to achieve good results over 100 epochs. The training was conducted in a TPU v4 environment provided by Google Colab, and results might vary with different hardware configurations. In terms of prediction time, the model can reveal steganographic images of size 124x124 pixels in under 2 seconds per

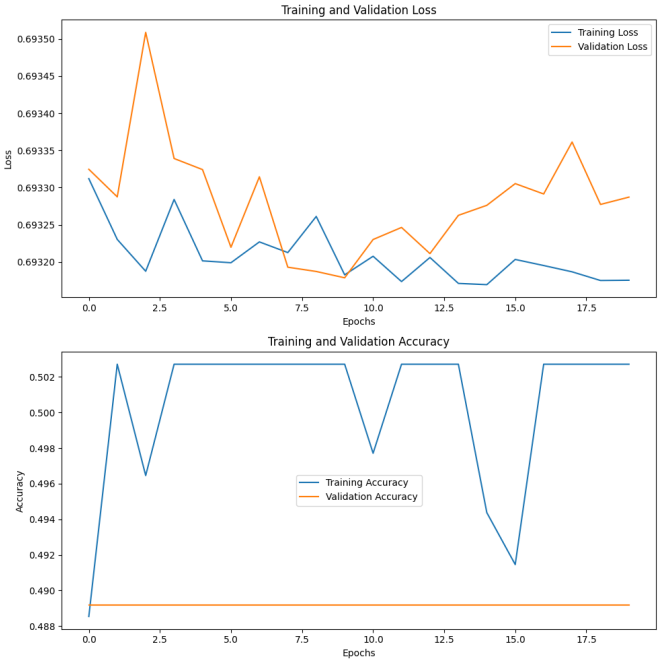


**Fig. 8.** Metrics Binary Classifier Based on Encoder-Weights

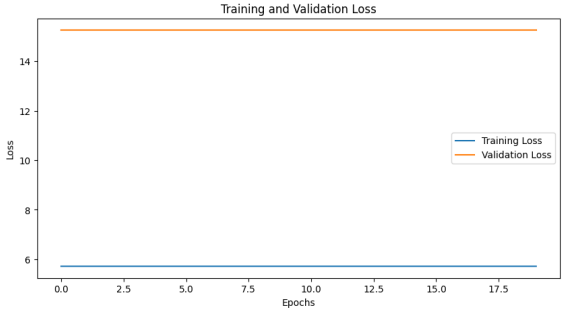


**Fig. 9.** Metrics Binary Classifier Based on Siamese Network

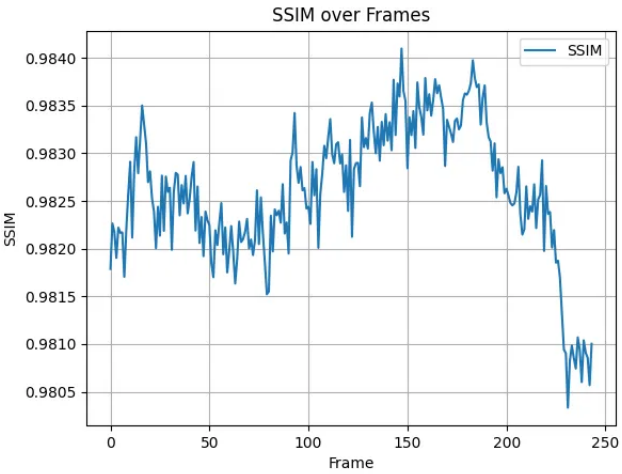


**Fig. 10.** Results for SSIM in the cover video.

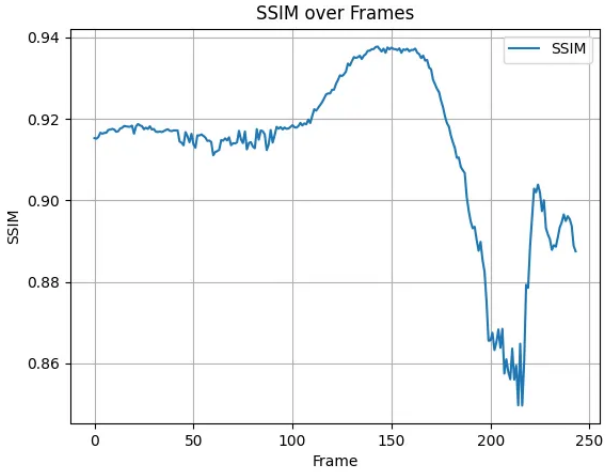


**Fig. 11.** Results for SSIM in the secret video.

image. However, these times can vary depending on the hardware used and potential quantization.

## Future Work

It is worth mentioning that during the experimentation period, the tool StegExpose was utilized. Created by Boehm (2014) [13], StegExpose is designed to detect whether an image contains a hidden text message using steganography. The tool evaluates the ability of models to evade detection through parameters such as Sample Pairs, Chi-Squared Attack, RS Analysis, and Primary Sets. However, this tool did not demonstrate any consistent results regarding its use in image-in-image steganography, and thus, it was not included in our investigation.

Similarly, our research into the state of the art in this field led us to a study conducted by Zhang, Cuesta-Infante, Xu, and Veeramachaneni (2019) [14]. They introduced an innovative concept called SteganoGAN, which leverages Generative Adversarial Networks. Their results were promising, indicating the potential of models to hide high-quality images in imperceptible ways. Therefore, SteganoGAN is an architecture worth exploring in future work to enhance the quality of videos subjected to video-in-video steganography.

Future research should focus on improving the fidelity of the recovered content and exploring advanced techniques for detecting steganographic content. Additionally, further investigation into the use of GANs, such as SteganoGAN, could lead to significant advancements in the field, enabling more effective and undetectable methods of video steganography.

## Conclusions

The results obtained from the experiments in video steganography demonstrate the effectiveness of using Convolutional Neural Networks (CNNs) for embedding and extracting hidden information in videos. The primary metric used for evaluation was the Structural Similarity Index Measure (SSIM), which indicated a high level of similarity between the cover and stego videos. The SSIM values for the cover video consistently remained above 0.9805, suggesting that the modifications introduced by the steganographic process were imperceptible to the human eye. This is a significant achievement, as it confirms the model's ability to conceal information without degrading the visual quality of the cover video.

For the secret video, the SSIM values ranged from 0.86 to 0.93. Although these values are slightly lower than those for the cover video, they still indicate a reasonable level of fidelity in the recovered content. This discrepancy suggests that while the model excels at embedding information, there is room for improvement in the extraction process to ensure higher fidelity of the recovered secret video.

The robustness of the model was further validated by the use of various data augmentation techniques during the training process, such as random rotations, flips, and adjustments to brightness and contrast. These techniques enhanced

the model's ability to generalize and perform well on diverse datasets, which is crucial for real-world applications.

In summary, the results of this study contribute significantly to the field of video steganography by demonstrating the viability of using CNNs for embedding and extracting hidden information. The high SSIM values achieved in both cover and secret videos affirm the model's capability to maintain visual quality while ensuring secure communication. However, further research is needed to improve the fidelity of the recovered content and to explore advanced techniques for detecting steganographic content.

When considering validation with other types of multimedia content, the challenge lies in finding a reliable metric to validate the training process. For instance, since this study focused on images, Mean Squared Error (MSE) per pixel was a suitable option. However, if the goal is to hide content like text, other metrics would need to be analyzed to ensure a good training process.

## References

1. Imagenet-a Dataset https://github.com/omar-vargas/imagenet-a-steganography-, last accessed 2024/06/25
2. Khan, A.A., et al.: IMG-forensics: Multimedia-enabled information hiding investigation using convolutional neural network. IET Image Process. 16, 2854–2862 (2022). https://doi.org/10.1049/ipr2.12272.
3. Kunhoth, J., Subramanian, N., Al-Maadeed, S. et al. Video steganography: recent advances and challenges. Multimed Tools Appl 82, 41943–41985 (2023). https://doi.org/10.1007/s11042-023-14844-w.
4. Liu, Y., Liu, S., Wang, Y., Zhao, H., Liu, S. (2019). Video steganography: A review. Neurocomputing, 335, 238-250.
5. AlKhodaidi, T., Gutub, A. Refining image steganography distribution for higher security multimedia counting-based secret-sharing. Multimed Tools Appl 80, 1143–1173 (2021). https://doi.org/10.1007/s11042-020-09720-w.
6. Wani, M. A., Sultan, B. (2023). Deep learning based image steganography: A review. WIREs Data Mining and Knowledge Discovery, 13(3), e1481. https://doi.org/10.1002/widm.1481
7. Zhang, Kevin Alex, et al. "SteganoGAN: High capacity image steganography with GANs." arXiv preprint arXiv:1901.03892 (2019).
8. Hashemi, S. H. O., Majidi, M. H., & Khorashadizadeh, S. (2022). Color Image steganography using Deep convolutional Autoencoders based on ResNet architecture. *arXiv preprint arXiv:2211.09409*.
9. Hayes, J., & Danezis, G. (2017). Generating steganographic images via adversarial training. Advances in neural information processing systems, 30.
10. Pevný, T., Filler, T., & Bas, P. (2010). Using high-dimensional image models to perform highly undetectable steganography. In Information Hiding: 12th International Conference, IH 2010, Calgary, AB, Canada, June 28-30, 2010, Revised Selected Papers 12 (pp. 161-177). Springer Berlin Heidelberg.
11. Wu, P., Yang, Y., & Li, X. (2018). Image-into-image steganography using deep convolutional network. In Advances in Multimedia Information Processing–PCM 2018: 19th Pacific-Rim Conference on Multimedia, Hefei, China, September 21-22, 2018, Proceedings, Part II 19 (pp. 792-802). Springer International Publishing.

12. Kunhoth, J., Subramanian, N., Al-Maadeed, S., & Bouridane, A. (2023). Video steganography: recent advances and challenges. Multimedia Tools and Applications, 82(27), 41943-41985.
13. Boehm, B. (2014). Stegexpose-A tool for detecting LSB steganography. *arXiv preprint arXiv:1410.6656*.
14. Zhang, K. A., Cuesta-Infante, A., Xu, L., & Veeramachaneni, K. (2019). SteganoGAN: High capacity image steganography with GANs. *arXiv preprint arXiv:1901.03892*.
15. DeepSteganography Model https://github.com/omar-vargas/model-deepsteganography/tree/main, last accessed 2024/06/25