ТЗ. №5(37)'2025 https://doi.org/10.32782/2786-9024/v3i5(37).344525

Інформаційна безпека

UDC 004.8:004.056

ENHANCED IMAGE STEGANOGRAPHY WITH KERAS-STEGANOGAN: A TENSORFLOW-BASED GAN

D.Yu. Khoma, Y.O. Bashkov
Postgraduate Student, Department of Applied Mathematics and Informatics
Ukraine Donetsk National Technical University, Donetsk, Ukraine

E-mail: dmytro.khoma@donntu.edu.ua

ABSTRACT

Image steganography is the process of embedding secret information within digital images such that the very presence of the message remains undetectable. Recent advances in deep learning, particularly in generative adversarial networks (GANs), have significantly improved both the payload capacity and perceptual quality of steganographic systems. The original SteganoGAN, implemented in Torch, achieved state-of-the-art performance by embedding up to 4.4 bits per pixel while maintaining strong resistance to steganalysis methods. However, the influence of the critic network on steganographic quality and learning stability remains insufficiently explored.

This paper presents Keras-SteganoGAN, a TensorFlow-based reimplementation and extension of SteganoGAN, designed to systematically analyze the role of the critic in adversarial steganographic training. Two variants of the model–one incorporating a critic and one without–were trained and compared across three encoder architectures: basic convolutional, residual, and dense. Each configuration was trained over five epochs with message depths ranging from 1 to 6 bits, allowing a comprehensive study of trade-offs between payload capacity, image distortion, and decoding accuracy.

Quantitative evaluation was conducted using standard image quality and steganographic metrics, including PSNR, SSIM, RS-BPP, and decoder accuracy. The results indicate that the inclusion of a critic improves perceptual quality and visual similarity at lower payloads, but its contribution diminishes as the message depth increases. These findings provide new insights into the interaction between encoder complexity, critic dynamics, and steganographic performance, offering guidance for the design of future GAN-based steganography systems.

Key words: *SteganoGAN, Keras, TensorFlow, image steganography, GAN, encoder, critic network, payload capacity, residual, dense, decoder, adversarial learning, hidden message, similarity metrics.*

Introduction

The art and science of image steganography revolve around embedding secret information within an image so that the presence of the hidden data remains imperceptible to observers. Unlike cryptography, where the primary goal is to secure the contents of the message by making it unreadable to adversaries, steganography goes a step further by ensuring that even the existence of the message is concealed. In typical usage scenarios, a sender encodes a secret message into a cover image, and transmits the image to a receiver who can extract the hidden data. This makes steganography particularly useful in situations where the transmission of encrypted messages might attract attention or raise suspicion.

The fundamental challenge of image steganography is to maximize the amount of data that can be hidden in an image without introducing visible artifacts or detectable anomalies. Steganography should ideally preserve the appearance of the cover image so that even sophisticated analysis tools or trained observers cannot detect alterations. However, this balance between embedding capacity and image quality has proven difficult to achieve. Traditional image steganography techniques, such as least significant bit (LSB) manipulation, can effectively hide small amounts of data but often introduce visible distortions when larger payloads are embedded. Moreover, automated steganalysis tools have become increasingly capable of detecting these modifications, limiting the effectiveness of these conventional approaches.

For a long time, traditional steganographic methods were only able to achieve modest payload capacities, typically up to around 0.4 bits per pixel (bpp) [1]. As payloads increase beyond this threshold, the chances of

Т3. №5(37)'2025 https://doi.org/10.32782/2786-9024/v3i5(37).344525

introducing detectable artifacts increase, and the cover image becomes more susceptible to analysis by automated steganalysis techniques. These tools can identify subtle inconsistencies in the image, such as irregularities in pixel distributions, that signal the presence of hidden data. In extreme cases, these distortions are noticeable to the human eye, rendering the steganographic technique ineffective for its purpose of secrecy.

In recent years, the rise of deep learning, particularly with the advent of neural networks, has significantly advanced the field of image steganography. Unlike traditional methods, which rely on predefined rules or statistical models to hide information, deep learning approaches have shown a remarkable ability to optimize both the payload capacity and image quality. These neural network-based methods are able to learn intricate patterns within images, making it possible to embed more data while minimizing detectable distortions. This has given rise to a new class of steganographic techniques that leverage generative models, such as generative adversarial networks (GANs), to create more efficient and effective steganographic systems.

Among these advancements is SteganoGAN, a Artificial Intelligence (AI) model developed by [2], which demonstrated the power of GANs in the realm of image steganography. SteganoGAN represents one of the first implementations of a fully end-to-end deep learning-based system for hiding arbitrary binary data in images, rather than simply embedding one image inside another. By employing a GAN architecture [3], SteganoGAN optimizes the perceptual quality of the generated steganographic images while simultaneously enhancing the embedding capacity. With the use of a critic network, the model is trained to produce steganographic images that are virtually indistinguishable from the original cover images, allowing it to evade detection by standard steganalysis tools. The original SteganoGAN achieved a payload capacity of up to 4.4 bits per pixel, a significant improvement over traditional methods.

However, despite these advancements, some aspects of SteganoGAN's design were left unexplored in the original work. One such area is the exact process by which the message tensor is created. In SteganoGAN, messages are encoded using the Reed-Solomon error correction code, which is designed to improve the reliability of message recovery by correcting errors in the decoded message. After encoding, the message is converted to bits and packed into a tensor, with message bits placed sequentially, divided by 32 zero bits. This encoding method improves the error-correction capabilities of the model, but it also adds complexity to the overall system.

In our work, we propose KerasSteganoGAN, a reimplementation of SteganoGAN in TensorFlow, with a key modification: the removal of Reed-Solomon encoding and decoding. By eliminating this step, we streamline the message embedding process while maintaining the core strengths of the GAN-based architecture. The motivation behind this change is to reduce computational overhead and complexity while still achieving effective message recovery. We also investigate the role of the critic network by introducing two versions of KerasSteganoGAN – one with a critic network and one without – allowing us to explore how the inclusion of the critic affects image quality and steganographic performance.

Research Objectives And Tasks

The primary objective of this research is to develop and evaluate Keras-SteganoGAN, a TensorFlow-based reimplementation of the original SteganoGAN model, designed to simplify the message embedding process while maintaining high image quality and embedding capacity. By removing the Reed-Solomon error correction mechanism, the study aims to reduce computational complexity and training overhead without compromising the accuracy of message extraction. This streamlined architecture seeks to demonstrate that robust message recovery and imperceptible image quality can still be achieved through careful network design and optimization within the GAN framework.

A secondary objective of this work is to analyze the influence of the critic network on the overall performance of the model. To this end, two distinct versions of KerasSteganoGAN are implemented: one including a critic component and another operating without it. The comparative analysis between these variants focuses on evaluating their impact on key performance metrics such as payload capacity, decoding accuracy, Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index Measure (SSIM). This comparison allows for a deeper understanding of how adversarial learning contributes to the balance between embedding fidelity and visual indistinguishability.

Finally, the research aims to empirically validate the effectiveness of KerasSteganoGAN through extensive experiments on benchmark image datasets. The study's tasks include designing and training the models, measuring and comparing their quantitative and qualitative performance, and analyzing trade-offs between computational efficiency and steganographic robustness. The results are expected to provide insights into optimizing GAN-based steganographic architectures for practical use, setting the groundwork for further exploration of adaptive and lightweight deep learning techniques in the field of image steganography.

Research Materials And Methods

SteganoGAN is a groundbreaking that leverages the power of GANs to tackle the challenges of image steganography. Traditional approaches to image

ISSN 2786-9024

ТЗ. №5(37)'2025 https://doi.org/10.32782/2786-9024/v3i5(37).344525

steganography often suffer from limited payload capacity and the risk of introducing visible artifacts that can be detected by automated steganalysis tools. By using a GAN-based architecture, SteganoGAN overcomes these limitations, significantly enhancing the embedding rate while maintaining high image quality [4].

At its core, SteganoGAN is an end-to-end deep learning model that allows for the hiding of arbitrary binary data inside images, written with Python library called PyTorch. The model's architecture consists of three main components: an encoder, a decoder, and a critic network.

The encoder is responsible for embedding the secret message into the cover image, transforming it into a steganographic image. The decoder, on the other hand, works to recover the hidden message from the steganographic image. The critic network plays the role of a discriminator in the GAN framework, evaluating how close the generated steganographic image is to the original cover image [5]. This adversarial relationship between the generator (encoder) and the critic enables the model to produce steganographic images that are nearly indistinguishable from the original ones, making it difficult for steganalysis tools to detect the presence of hidden data.

One of the key innovations in SteganoGAN is the use of multiple loss functions to optimize the encoder, decoder, and critic simultaneously. By balancing these losses – specifically, the decoding accuracy, the perceptual similarity between the cover and steganographic images, and the realism of the generated image – the model is able to achieve a high payload capacity while maintaining image fidelity.

The architecture of SteganoGAN is carefully designed to balance embedding capacity and image quality. It includes three key components: the encoder, the decoder, and the critic network, which together work within an adversarial training framework and shown on Figure 1. This allows SteganoGAN to hide arbitrary binary data in images while maintaining their visual integrity, like in [6].

Encoder Network

SteganoGAN incorporates three different encoder architectures, each designed to explore different methods of embedding messages into images: the basic encoder, the residual encoder, and the dense encoder. Each of these architectures (Figure 2) handles the feature extraction and message embedding processes differently, which impacts both the quality of the generated steganographic image and the model's ability to recover the hidden message.

Basic Encoder – the the simplest variant, designed with a straightforward convolutional architecture. It starts by processing the cover image through several convolutional layers, which transform the image into a feature map. The binary message is then concatenated to these features and passed through additional convolutional layers. The final output is a steganographic image that contains the embedded message. This architecture is relatively simple and fast to train, but its limitation is that it may not efficiently capture intricate image details or provide the best possible image quality when embedding large amounts of data. The lack of skip connections or advanced feature reuse mechanisms means that the model might struggle with higher message depths or complex images.

Residual Encoder – builds upon the basic encoder by introducing residual connections, a technique originally popularized by ResNet architectures [7]. In this design, after the message is concatenated to the image features and processed through the convolutional layers, the output of the encoder is added to the original cover image.

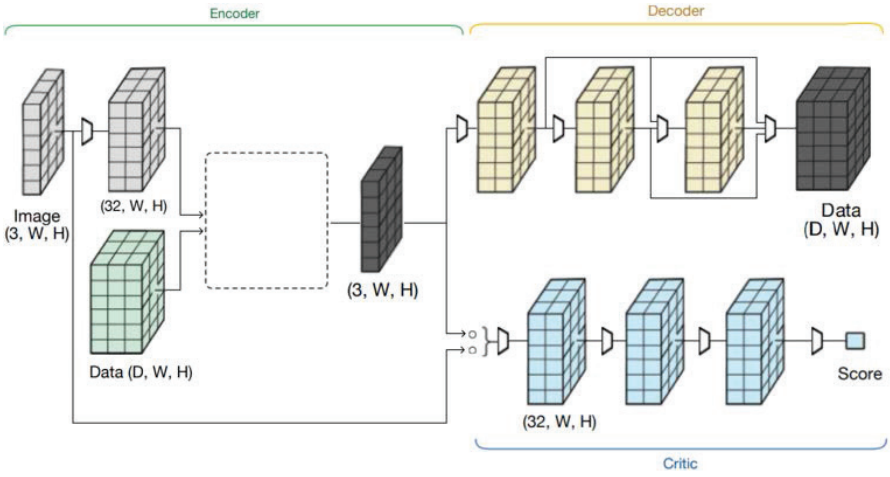

Fig. 1. Original SteganoGAN model architecture

ТЗ. №5(37)'2025 https://doi.org/10.32782/2786-9024/v3i5(37).344525

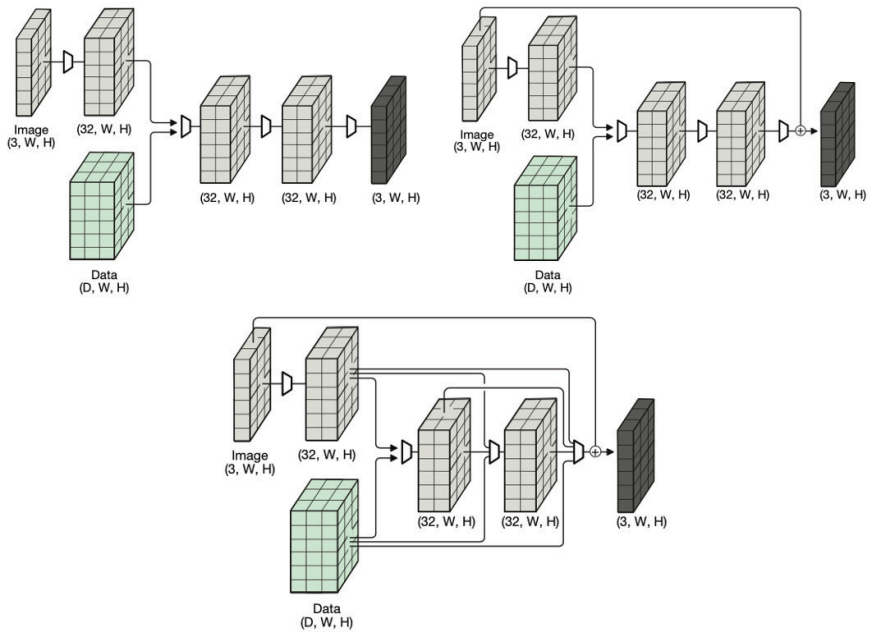

Fig. 2. Encoder different types (Basic, Residual and Dense)

This means that the encoder is learning to produce a residual image – an image that represents the difference between the cover image and the steganographic image – rather than the steganographic image itself. The advantage of using residual connections is that they help prevent vanishing gradient problems during training and enable the model to focus on the minimal changes needed to embed the message. By only adjusting parts of the image where necessary, the residual encoder tends to produce higher-quality steganographic images, especially when dealing with larger payloads. This architecture also typically converges faster than the basic encoder.

Dense Encoder – the most advanced of the three and is inspired by the DenseNet architecture [8]. In this design, the output of each convolutional layer is densely connected to the inputs of all subsequent layers. In other words, the feature maps produced by each layer are concatenated with the feature maps of all previous layers, ensuring that the encoder reuses features across the network. This results in feature reuse, meaning that the model can more effectively capture complex image details and use the most relevant information for embedding the message. Dense connections mitigate the vanishing gradient problem more effectively than residual connections and encourage the model to leverage both low-level and high-level features simultaneously. This makes the dense encoder particularly well-suited for scenarios where the payload is large or the image content is complex. However, the trade-off is that this architecture is more computationally expensive and requires more memory than the basic or residual encoders.

Decoder Network

In SteganoGAN, two decoder architectures are employed: the basic decoder and the dense decoder. The decoder's primary role is to extract the hidden message from the steganographic image, and the effectiveness of this process is crucial for ensuring accurate message recovery. The architecture of the decoder can significantly influence how well it performs under different conditions, particularly when handling varying payload sizes and complex image content.

Basic Decoder – follows a straightforward approach, much like the basic encoder. It consists of several convolutional layers that transform the steganographic image back into a feature representation. These features are then processed to recover the original binary message. Just like Basic Endoder, the design of the basic decoder is simple, with no advanced connectivity mechanisms between layers. While the basic decoder is efficient and works well for smaller payloads, its simplicity can become a limiting factor when dealing with larger message depths or more complex steganographic images [9-10]. It processes each layer sequentially, without taking advantage of previous layer outputs, which can lead to less accurate message recovery in more challenging scenarios. Should be mentioned that Basic Decoder only uses with Basic or Residual Encoder.

Dense Decoder – builds upon the same principle as the dense encoder, utilizing dense connections to enhance feature extraction and reuse. In this architecture, each layer's output is concatenated with the outputs of all preceding layers. This ensures that every convolutional layer has access to the full set of features

Т3. №5(37)'2025 https://doi.org/10.32782/2786-9024/v3i5(37).344525

extracted by earlier layers, allowing the decoder to leverage both low-level and high-level information simultaneously.

The dense decoder is particularly well-suited for recovering messages from images where the message depth is high or the cover image is complex. By maintaining access to all previous feature maps, the dense decoder can more effectively reconstruct the binary message. This architecture excels in scenarios where high accuracy is essential, as it mitigates the risk of losing critical information during the decoding process.

Critic Network

The critic network is crucial to the adversarial training process. It distinguishes between real images (the original cover images) and fake images (the steganographic images generated by the encoder). The critic's feedback guides the encoder, pushing it to generate steganographic images that look more realistic and are harder to distinguish from the original cover images. The critic uses the Wasserstein loss, which is designed to improve the training stability of GANs and ensures that the generated images closely resemble real cover images [11-13].

Loss Functions

SteganoGAN employs three primary loss functions to optimize the encoder, decoder, and critic simultaneously:

– Decoder loss – measures how well the decoder D recovers the hidden message in steganographic message $\mathcal{E}(X, M)$, where X is original image, and M is original message. It uses binary cross-entropy loss to compare the original message to the decoded message $D(\mathcal{E}(X, M))$. The goal is to minimize the error between the original and the recovered with decoder binary messages.

$$\mathcal{L}_d = \mathbb{E}_{X \sim \mathbb{P}_C} CrossEntorpy\left(D\left(\mathcal{E}(X, M)\right), M\right) \quad (1)$$

– Image similarity loss – this loss ensures that the steganographic image remains visually similar to the original cover image. The image similarity loss is calculated using the mean square error (MSE), which measures the difference between each pixel in the cover image and the corresponding pixel in the steganographic image. Minimizing this error helps preserve the appearance of the cover image.

$$\mathcal{L}_s = \mathbb{E}_{X \sim \mathbb{P}_C} \frac{1}{3 \times W \times H} X - \mathcal{E}(X, M)_2^2 \quad (2)$$

– Realness loss – returns critic C score of steganographic image.

$$\mathcal{L}_r = \mathbb{E}_{X \sim \mathbb{P}_C} C\left(\mathcal{E}(X, M)\right) \quad (3)$$

The total loss is the sum of these three losses: the decoder loss, the image similarity loss, and the realness loss.

$$minimize \mathcal{L}_d + \mathcal{L}_s + \mathcal{L}_r \quad (4)$$

By minimizing this combined loss, SteganoGAN optimizes the quality of the steganographic images while ensuring reliable message recovery. SteganoGAN uses this combined loss to train both encoder and decoder model weights.

Critic loss – evaluates how realistic the steganographic images are compared to the original cover images.

$$\mathcal{L}_c = \mathbb{E}_{X \sim \mathbb{P}_C} C(X) - \mathbb{E}_{X \sim \mathbb{P}_C} C\left(\mathcal{E}(X, M)\right) \quad (5)$$

The critic loss is based on the Wasserstein distance, which compares the distributions of the real images (cover images) and the generated images (steganographic images). The encoder aims to generate images that minimize the critic's ability to distinguish between the real and fake images.

Keras Steganogan Architecture

KerasSteganoGAN is a TensorFlow-based reimplementa-tion of the original SteganoGAN architecture with several notable modifications to simplify the message encoding and decoding processes while maintaining the overall structure and functionality of the original model. Instead of modifying original PyTotch-based StegnoGAN, KerasSteganoGAN was rewritten with more functional and popular Python library called TensorFlow. First reason to make TensorFlow-based SteganoGAN is possibility of using it only with definite version of PyTorch library and Python. Another reason is possibility of using LiteRT library of making lightweighted mobile versions of AI models.

Key modification in KerasSteganoGAN is the removal of the Reed-Solomon error correction method, which was previously used in SteganoGAN to enhance message recovery accuracy. In SteganoGAN, the binary message was first encoded with Reed-Solomon, converted into bits, and then placed into a tensor, separated by blocks of zero bits. This process added computational complexity while improving error tolerance during decoding.

In KerasSteganoGAN, we opted to remove the Reed-Solomon encoding and decoding steps, simplifying the model's structure. By doing so, we rely solely on the AI models (encoder and decoder) to manage the embedding and extraction of messages. This change reduces overhead and streamlines the overall pipeline without significantly impacting performance. The hidden message is now directly embedded into the cover image, and its recovery is entirely dependent on the performance of the neural network models.

Despite these modifications, the core components of the architecture remain intact, including the encoder-decoder structure and the option to utilize different types of encoders (basic, residual, and dense). The removal of Reed-Solomon encoding enables the model to focus purely on leveraging the strengths of the neural network for accurate message recovery, which simplifies

ТЗ. №5(37)'2025 https://doi.org/10.32782/2786-9024/v3i5(37).344525

the training and deployment processes while maintaining high payload capacity and image quality.

Another significant change that contains KerasSteganoGAN is that we use Sigmoid activation function at the last convolution layers of each encoder and decoder models in order to return data in a range of [-1, 1]. In the case of the encoder, returned data will be converted to an image, or in the case of the decoder, it will be converted to a tensor of binary data. In the original SteganoGAN model, both encoder and decoder models have no activation functions at the last convolution layers [14-15].

KerasSteganoGAN architecture allows generation of steganographic images with the size of 128x128 pixels, though SteganoGAN allows generation of stego-images with original size of cover-images. The reason of this is the difficulty of the implementation of such an architecture as well as the complexity of training.

One of the key explorations in KerasSteganoGAN is the introduction of two distinct model variants: with a critic and without a critic. In the original SteganoGAN, the critic network played an important role in the adversarial training process by helping the model learn to generate steganographic images that are indistinguishable from real images. The critic evaluates how realistic the generated steganographic images are and provides feedback to improve the encoder's output.

In KerasSteganoGAN, we decided to investigate the specific impact of the critic network on the quality of the steganographic images and the accuracy of message recovery. To do so, we created two versions of the model:

- One version includes the critic network, mimicking the adversarial setup from SteganoGAN.
- The other version operates without a critic network, meaning the encoder and decoder are trained without the additional adversarial feedback.

The motivation for this comparison is to analyze whether the critic network contributes significantly to improving the visual quality of the images and the reliability of message recovery or if a non-adversarial approach can yield comparable results. Both variants still use the same encoder-decoder structures, with the option to choose between basic, residual, and dense encoder types, but the inclusion or exclusion of the critic alters the training dynamics. Figure 3 presents KerasSteganoGAN architecture without critic with basic or dense decoder types while encoder types could be the same as at the original SteganoGAN: basic, residual or dense.

Training both model variants allows us to compare the influence of adversarial feedback in the GAN framework versus simpler encoder-decoder setups. This exploration is aimed at understanding whether the critic's complexity justifies its potential benefits, especially in terms of computational cost and training time.

Training And Comparision

To thoroughly evaluate the performance of KerasSteganoGAN models with and without the critic network, we used a set of key metrics that assess both the quality of the steganographic images and the effectiveness of message recovery. These metrics – RS-BPP (Reed-Solomon bits per pixel), PSNR (Peak Signal-to-Noise Ratio), and SSIM (Structural Similarity Index Measure) – are standard in the field of steganography, providing insights

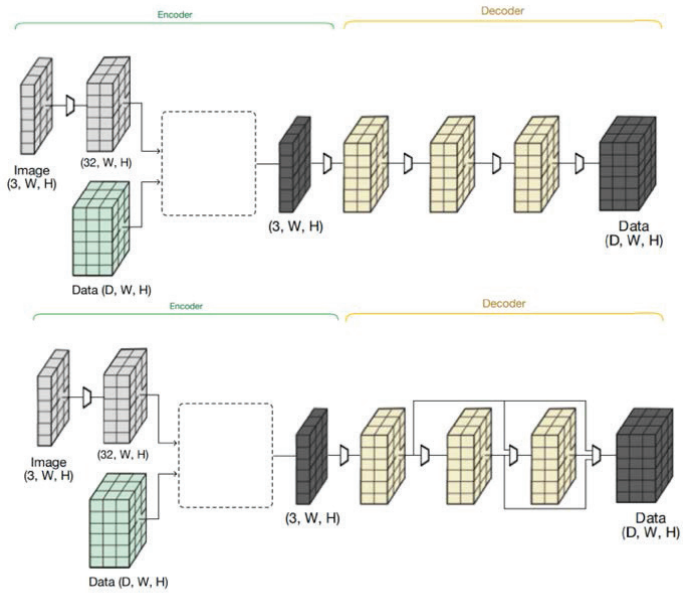

Fig. 3. Critic-less KerasSteganoGAN architecture with basic or dense decoder types

ТЗ. №5(37)'2025 https://doi.org/10.32782/2786-9024/v3i5(37).344525

into the image quality, message capacity, and accuracy of the decoder. Detailed description of these metrics is provided in [16].

To train our model Div2K [17] dataset was used with batch size of 4. The train dataset consists of 700 pictures of different sizes, while the validation dataset consists of only 100 pictures. Due to the absence of GPU with CUDA cores, our neural network training was done on CPU Intel i9, and took 15 minutes to train the model on 5 epochs.

PSNR measures the difference in quality between the steganographic image and the original cover image. A higher PSNR value indicates that the two images are more similar, with fewer distortions or artifacts in the steganographic image. In the context of KerasSteganoGAN, we use PSNR to evaluate how well the model preserves the visual quality of the cover image after embedding the hidden message. The influence of the critic network is particularly highlighted by this metric, as models with a critic are expected to generate steganographic images that closely resemble the cover images.

SSIM is another metric used to evaluate the quality of the steganographic image, focusing on the perceptual similarity between the cover and stego-images. It measures structural changes, taking into account luminance, contrast, and texture. Like PSNR, higher SSIM values indicate better preservation of the image structure. To visualize the impact of the critic network on image quality, plots of SSIM for different encoder types and data depths will demonstrate how the critic influences the visual similarity between the stego and cover images.

The RS-BPP metric calculates the effective number of bits per pixel that can be reliably hidden and recovered in the image. This metric is derived from the decoder accuracy and the message depth. Although we removed the Reed-Solomon encoding/decoding process in KerasSteganoGAN, we still use the RS-BPP formula to estimate the capacity of the model. It provides a clear measure of how efficiently the model can embed and recover data, particularly when comparing different encoder types and their ability to handle various message depths.

These three metrics allow us to assess the performance of KerasSteganoGAN from different perspectives: PSNR and SSIM evaluate the visual fidelity of the images, while RS-BPP measures the model's capacity to embed and recover data effectively. The upcoming sections will illustrate these metrics through various plots, particularly focusing on how the inclusion of the critic network affects image quality and message recovery.

To comprehensively compare the performance of KerasSteganoGAN models with and without a critic network, we trained both variants using three encoder-decoder configurations:

- basic encoder – basic decoder;
- residual encoder – basic decoder;
- dense encoder – dense decoder.

For each of these encoder-decoder pairs, we experimented with six different message depths (1 to 6 bits per pixel), varying the amount of data embedded in each image. Each configuration was trained for 5 epochs, as this was sufficient to observe the general trends in the models' performance while keeping the training time manageable. The impact of these choices on the performance of the models is reflected in several metrics – PSNR, SSIM, and RS-BPP – as shown in the figures that follow.

In Figure 4, we present PSNR results for all three encoder configurations (basic, residual, and dense) across different message depths. Here, it becomes evident that models with a critic network generally achieve higher PSNR values, particularly at higher data depths, indicating better preservation of image quality. The residual encoder, paired with a basic decoder, shows a significant improvement in PSNR compared to the basic

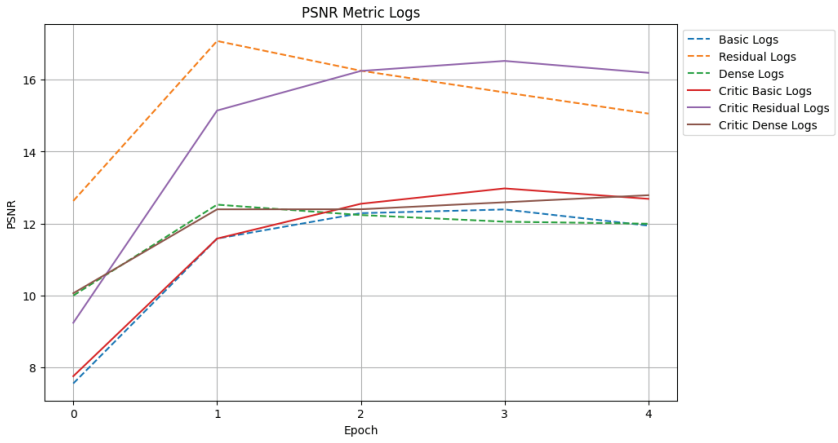

Fig. 4. Plot of SSIM metrics during 5 epochs of training KerasSteganoGAN with and without critic model

Т3. №5(37)'2025 https://doi.org/10.32782/2786-9024/v3i5(37).344525

encoder, especially when trained with a critic. The dense encoder achieves the best overall results, with PSNR values staying high even with 6 bits per pixel of data, highlighting its superior ability to handle larger payloads.

Similarly, Figure 5 showcases the SSIM metric for the same encoder-decoder configurations. The dense encoder consistently achieves higher SSIM values, indicating a better structural similarity between the stego and cover images. Models with a critic tend to perform better in terms of SSIM across all data depths, reinforcing the idea that the critic helps the encoder preserve image structure more effectively.

The RS-BPP metric (shown in Figure 6) provides insight into the effective payload capacity of the models.

Here, the dense encoder again outperforms the basic and residual encoders, particularly at higher message depths. As expected, models with a critic achieve slightly higher RS-BPP values, reflecting their ability to embed and recover data more accurately. However, the improvement in RS-BPP is not as dramatic as the gains seen in PSNR and SSIM, suggesting that the critic's primary benefit lies in improving image quality rather than significantly boosting payload capacity.

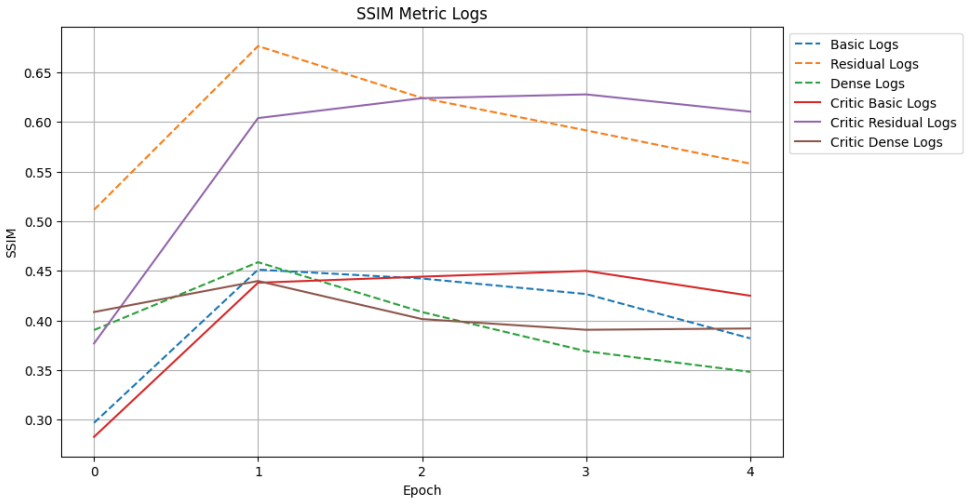

Fig. 5. Plot of PSNR metrics during 5 epochs of training KerasSteganoGAN with and without critic model

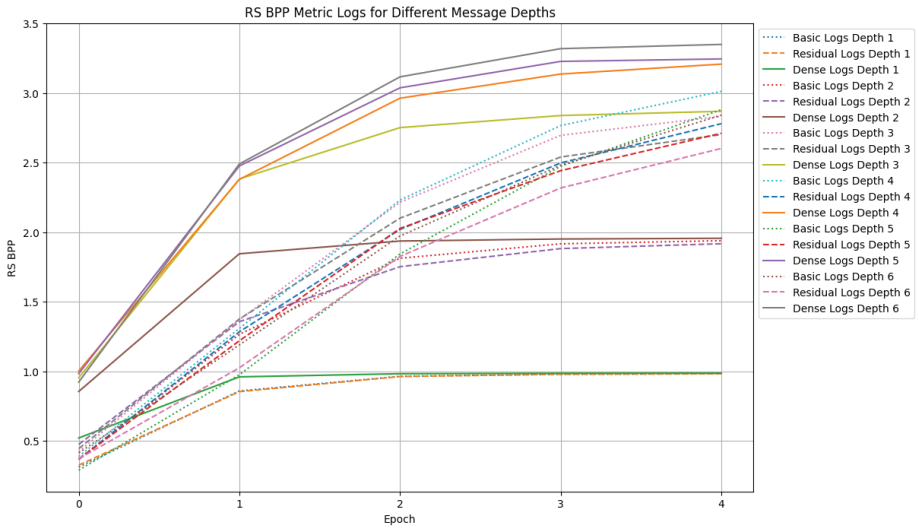

Fig. 6. RS-BPP metric during training KerasSteganoGAN with different payloads for all types of Encoder model

ТЗ. №5(37)'2025 https://doi.org/10.32782/2786-9024/v3i5(37).344525

Research Experiments And Results

These results highlight the consistent pattern observed across metrics – models with a critic tend to produce higher quality images, as evidenced by their superior PSNR and SSIM values, while dense encoder architectures excel in both image quality and data embedding efficiency.

To better understand the effectiveness of our trained models, there is table 1, which contains Decoder accuracy, PSNR, SSIM, and RS-BPP metrics for each KerasSteganoGAN model with different data depth, encoder architecture, and critic model presence. Figure 7 presents the original image in the first row, the stego-image created by the KerasSteganoGAN model with critic, dense encoder, and data-depth 6 trained on 5 epochs on the second row, and their difference in the gray-scale on the third row.

The decoder accuracy remains high across all models and data depths, particularly for smaller message depths (1 and 2 bits per pixel). Both models with and without a critic achieve nearly perfect accuracy at lower message depths, with a slight drop as the data depth increases. At data depth 6, the model with a critic shows slightly better accuracy for the dense encoder (77%) compared to the non-critic model (78%), indicating the critic's positive impact in higher payload scenarios.

RS-BPP, which measures the effective payload, shows a consistent trend: the dense encoder performs best across all data depths. For instance, at data depth 6, the dense encoder achieves an RS-BPP of 3.34 (with critic) and 3.42 (without critic), outperforming both the basic and residual encoders. The basic encoder shows relatively lower RS-BPP across all configurations, with values ranging from 0.98 at depth 1 to 2.84 at depth 6 (with a critic). In general, models without a critic show slightly higher RS-BPP values, but the difference is not significant.

Table 1. Metrics For Different Model Architecture And Payload Depth

Critic	D	Decoder Accuracy			RS-BPP			PSNR			SSIM		
		Basic	Residual	Dense	Basic	Residual	Dense	Basic	Residual	Dense	Basic	Residual	Dense
Yes	1	0.99	0.99	0.99	0.98	0.98	0.98	13.11	16.69	14.44	0.47	0.63	0.52
	2	0.98	0.97	0.98	1.93	1.91	1.95	12.13	14.66	13.93	0.39	0.53	0.49
	3	0.97	0.95	0.97	2.83	2.70	2.86	10.81	12.20	11.60	0.26	0.36	0.29
	4	0.87	0.84	0.90	3.01	2.77	3.20	11.21	13.11	11.55	0.29	0.42	0.28
	5	0.78	0.77	0.82	2.87	2.71	3.24	11.99	14.95	12.08	0.36	0.54	0.33
	6	0.73	0.71	0.77	2.84	2.60	3.34	12.68	16.19	12.78	0.42	0.61	0.39
No	1	0.99	0.99	0.99	0.98	0.98	0.99	12.34	15.62	14.82	0.42	0.59	0.55
	2	0.99	0.99	0.99	1.96	1.96	1.96	11.62	14.06	13.12	0.35	0.50	0.45
	3	0.97	0.95	0.98	2.82	2.74	2.90	10.78	11.55	10.99	0.26	0.31	0.27
	4	0.88	0.83	0.90	3.08	2.65	3.22	10.66	12.93	10.80	0.25	0.43	0.24
	5	0.79	0.78	0.83	2.97	2.84	3.37	11.64	13.62	11.60	0.35	0.47	0.30
	6	0.75	0.73	0.78	3.02	2.79	3.42	11.94	15.05	11.99	0.38	0.55	0.34

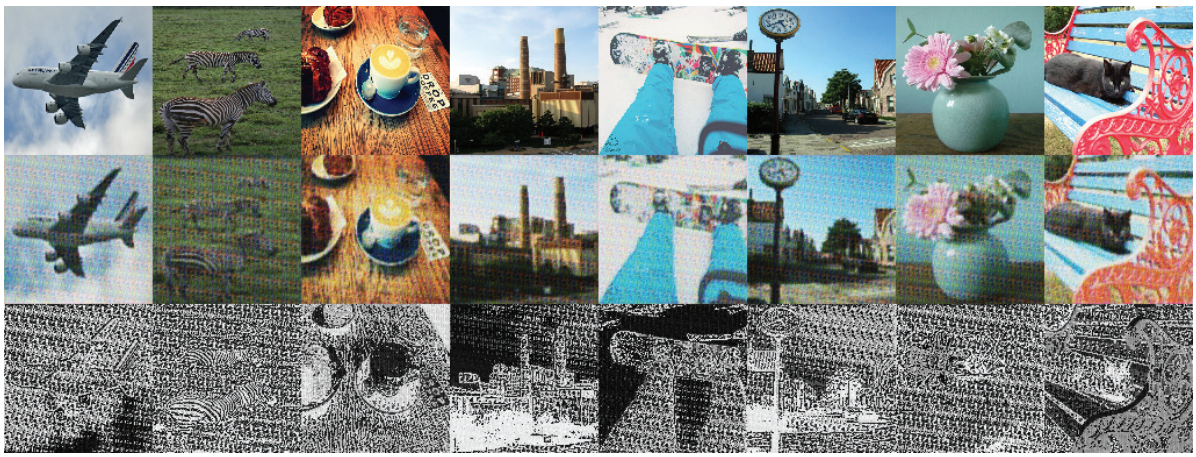

Fig. 7. Original resized, callback, their difference images on 5 epochs of the Dense model with critic and data-depth equals 6

ТЗ. №5(37)'2025 https://doi.org/10.32782/2786-9024/v3i5(37).344525

In terms of PSNR, the dense encoder consistently performs better across all message depths compared to the basic and residual encoders. For example, at data depth 6, the dense encoder achieves a PSNR of 12.78 (with critic) and 11.99 (without critic), outperforming the other two encoders. The critic-enhanced models tend to have better PSNR values at higher data depths. At data depth 6, the basic encoder with a critic achieves a PSNR of 12.68 compared to 11.94 without a critic. This indicates that the critic helps preserve image quality as more data is embedded into the image.

SSIM results follow a similar pattern to PSNR, with the dense encoder showing the best performance across all message depths. For example, at data depth 6, the dense encoder achieves an SSIM of 0.39 (with critic) and 0.34 (without critic). The critic-enhanced models consistently achieve higher SSIM values across all configurations. At data depth 6, the residual encoder with a critic scores 0.61, compared to 0.55 without a critic, reinforcing the idea that the critic network helps maintain the structural integrity of the image. The residual encoder benefits the most from the critic in terms of SSIM. For instance, at data depth 6, the residual encoder with a critic scores 0.61, significantly higher than the 0.55 scored by the non-critic model.

Dense encoder consistently outperforms both basic and residual encoders across all metrics, showing its strength in handling larger payloads (higher data depths). Models with a critic generally perform better in terms of PSNR and SSIM, indicating that the critic helps improve image quality and structural similarity, especially at higher data depths. The non-critic models show slightly higher RS-BPP values, but the difference is not substantial, suggesting that while the critic improves image quality, it does not greatly impact the payload capacity.

Conclusion

In this paper, we presented KerasSteganoGAN, a TensorFlow-based reimplementation and extension of the original SteganoGAN model, with key modifications, including the removal of the Reed-Solomon encoding/decoding process and the introduction of two distinct model variants: with and without a critic network. Our goal was to assess the impact of these changes on the model's ability to hide and recover messages while maintaining the quality of the steganographic images.

Through a comprehensive evaluation using RS-BPP, PSNR, and SSIM metrics across different encoder-decoder architectures and message depths (ranging from 1 to 6 bits per pixel), we observed several important trends:

- Dense encoders, regardless of whether a critic was used, consistently outperformed both basic and residual encoders in terms of both image quality and data embedding efficiency. Dense connections allowed for better feature reuse, resulting in higher RS-BPP, PSNR, and SSIM values across all configurations.
- The critic network played a crucial role in improving image quality, particularly at higher message depths. Models with a critic achieved higher PSNR and SSIM values, indicating that the critic helped preserve both the visual and structural integrity of the steganographic images.
- While models without a critic achieved slightly higher RS-BPP values, the improvement was marginal, suggesting that the critic network primarily enhances image quality rather than significantly increasing payload capacity.

Overall, our results suggest that using a dense encoder with a critic network yields the best trade-off between payload capacity and image quality, making it an ideal choice for high-capacity steganographic tasks. However, given that all models were only trained for 5 epochs, our results also indicate that more extensive training (at least 32 epochs, as used in the original SteganoGAN) would likely lead to further improvements in performance, especially for more complex encoder architectures like the dense encoder.

Future work will focus on exploring additional encoder architectures and loss functions to further optimize the performance and enhance its applicability in real-world steganography scenarios. We plan to retrain the model and perform a comparative analysis under equivalent training conditions, such as training duration, to ensure a fair evaluation of the model's improvements. Additionally, we aim to generate images at their original size to better reflect practical use cases and assess the model's performance with higher-resolution data.

Conflict of Interest

The authors declare that they have no conflict of interest regarding this study, including financial, personal, authorship-related, or any other type of conflict that could have influenced the research or its results presented in this article.

Funding

This research was conducted without any financial support.

Data Availability

This manuscript has no associated datasets.

ПОКРАЩЕНА СТЕГАНОГРАФІЯ ЗОБРАЖЕНЬ ЗА ДОПОМОГОЮ KERAS-STEGANOGAN: GAN НА ОСНОВІ TENSORFLOW

Дмитро Хома, Євген Башков

Стеганографія зображень – це процес вбудовування секретної інформації в цифрові зображення таким чином, щоб сам факт існування

ТЗ. №5(37)'2025 https://doi.org/10.32782/2786-9024/v3i5(37).344525

повідомлення залишався непомітним. Останні досягнення у сфері глибокого навчання, зокрема в генеративно-змагальних мережах (GAN), суттєво покращили як пропускну здатність стеганографічних систем, так і якість відтворення зображень. Оригінальна модель SteganoGAN, реалізована у Torch, досягла найсучасніших результатів, забезпечуючи приховування до 4.4 біт на піксель при високій стійкості до методів стегоаналізу. Проте вплив критика (critic network) на якість стеганографії та стабільність навчання досі залишається недостатньо вивченим.

У цій роботі представлено Keras-SteganoGAN – реалізацію та розширення SteganoGAN на базі TensorFlow, призначену для систематичного аналізу ролі критика в процесі змагального стеганографічного навчання. Було розроблено та порівняно дві версії моделі – з критиком і без нього – на основі трьох архітектур енкодера: basic, residual та dense. Кожну конфігурацію навчали протягом п'яти епох із глибиною повідомлення від 1 до 6 біт, що дало змогу всебічно дослідити компроміси між ємністю прихованих даних, спотворенням зображення та точністю декодування.

Кількісна оцінка виконувалася за допомогою стандартних метрик якості зображень і стеганографічної ефективності, включаючи PSNR, SSIM, RS-BPP та точність декодування. Результати показують, що наявність критика підвищує візуальну якість та схожість зображень за невеликих навантажень, проте його вплив зменшується зі збільшенням глибини повідомлення. Отримані результати надають нове розуміння взаємодії між складністю енкодера, динамікою критика та стеганографічною ефективністю, а також формують підґрунтя для подальшого вдосконалення систем стеганографії на базі GAN.

Ключові слова: *SteganoGAN, Keras, TensorFlow, стеганографія зображень, GAN, енкодер, критик, пропускна здатність, residual, dense, декодер, змагальне навчання, приховане повідомлення, метрики схожості.*

REFERENCES

[1] Pevnýˊ, T., Filler, T., and Bas, P. "Using high-dimensional image models to perform highly undetectable steganography. Information Hiding", 2010.

[2] Kevin Alex Zhang, Alfredo Cuesta-Infante, Lei Xu, and Kalyan Veeramachaneni, "Steganogan: High capacity image steganography with gans," arXiv preprint arXiv:1901.03892, 2019.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets. Advances in neural information processing systems", 27, 2014.

[4] Hayes, J. and Danezis, G. "Generating steganographic images via adversarial training". In NIPS, 2017.

[5] Baluja, S. "Hiding images in plain sight: Deep steganography". In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), Advances in Neural Information Processing Systems 30, pp. 2069–2079. Curran Associates, Inc., 2017.

[6] Zhu, J., Kaplan, R., Johnson, J., and Fei-Fei, L. "HiDDeN: Hiding data with deep networks". CoRR, abs/1807.09937, 2018.

[7] He, K., Zhang, X., Ren, S., and Sun, J. "Deep residual learning for image recognition". IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 770–778, 2016.

[8] Huang, G., Liu, Z., van der Maaten, L., and Weinberger, K. Q. "Densely connected convolutional networks". IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 2261–2269, 2017.

[9] Shao-Ping Lu, Rong Wang, Tao Zhong, and Paul L Rosin, "Large-capacity image steganography based on invertible neural networks", in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 10816– 10825.

[10] Donghui Hu, Liang Wang, Wenjie Jiang, Shuli Zheng, and Bin Li, "A novel image steganography method via deep convolutional generative adversarial networks", IEEE access, vol. 6, pp. 38303–38314, 2018.

[11] S. L. Kryvyi, Y. V. Boyko, S. D. Pogorilyy, O. F. Boretskyi, M. M. Glybovets. "Design of Grid Structures on the Basis of Transition Systems with the Substantiation of the Correctness of Their Operation". Cybernetics and Systems Analysis. January 2017, Volume 53, Issue 1, pp 105–114. Springer Science+Business Media New York 2017. https://doi.org/10.1007/s10559-017-9911-0

[12] S. D. Pogorilyy and I. Yu. Shkulipa. "A Conception for Creating a System of Parametric Design of Parallel Algorithms and Their Software Implementations". Cybernetics and System Analysis, Volume 45, Issue 6 (November 2009), p.p. 952-958. Springer Science and Business Media Inc. ISSN: 1060-0396.

[13] Tang, W., Tan, S., Li, B., and Huang, J. "Automatic steganographic distortion learning using a generative adversarial network". IEEE Signal Processing Letters, 24(10):1547–1551, Oct 2017. ISSN 1070-9908. doi: 10.1109/LSP.2017.2745572.

[14] Zhuo Zhang, Jia Liu, Yan Ke, Yu Lei, Jun Li, Minqing Zhang, and Xiaoyuan Yang, "Generative steganography by sampling," IEEE access, vol. 7, pp. 118586–118597, 2019.

[15] B Delina, "Information hiding: A new approach in text steganography," in Proceedings of the International Conference on Applied Computer and Applied Computational Science, World Scientific and Engineering Academy and Society (WSEAS 2008), 2008, pp. 689–695.

Т3. №5(37)’2025 https://doi.org/10.32782/2786-9024/v3i5(37).344525

[16] Khoma D.Yu., “Steganographic algorithms and stegoanalysis based on classical methods and neural networks” – Scientific works of DonNTU, series "Informatics, Cybernetics and Computing", No. 2 (37), 2024, p. 42–53. https://doi.org/10.31474/1996-1588-2023-2-37-42-53

[17] Agustsson, E. and Timofte, R. “NTIRE 2017 challenge on single image super-resolution: Dataset and study”. In The IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) Workshops, July 2017.

Стаття надійшла до редакції 30.09.2025
Стаття прийнята 12.10.2025
Статтю опубліковано 02.12.2025