Enhancing Steganography Security with Generative AI: A Robust Approach Using Content-Adaptive Techniques and FC_DenseNet

Ayyah Abdulhafidh Mahmoud Fadhl[1], Bander Ali Saleh Al-rimy[2], Sultan Ahmed Almalki[3]*, Tami Alghamdi[4], Azan Hamad Alkhorem[5], Frederick T. Sheldon[6]
Artificial Intelligence Department, Libyan International University, Benghazi, Libya[1]
School of Computing, University of Portsmouth, Portsmouth PO1 3HE, UK[2]
Computer Department-Applied College, Najran University, Najran 66462, Kingdom of Saudi Arabia[3]
Computer Science Department-Faculty of Computing and Information,
Al-Baha University, Al-Baha, 65779, Kingdom of Saudi Arabia[4]
Department of Computer Engineering-College of Computer Science and Information Technology,
Majmaah University, Al-Majmaah 11952, Kingdom of Saudi Arabia[5]
Department of Computer Science, University of Idaho, Moscow, ID 83844, USA[6]

***Abstract*—Content-adaptive image steganography based on minimizing the additive distortion function and Generative Adversarial Networks (GAN) is a promising trend. This approach can quickly generate an embedding probability map and has a higher security performance than hand-crafted methods. However, existing works have ignored the semantic information between neighbouring pixels and the NaN-loss scenarios, which leads to improper convergence. Such cases will degrade the generated Stego images' quality, decreasing the secret payload's security. FT_GAN performance, which incorporates feature reuse in generator architecture, has been investigated by proposing the FC_DenseNet-based generator herein. This investigation explores the superior semantic segmentation capabilities of FC_DenseNet, including feature reuse, implicit deep supervision, and the vanishing gradient problem alleviation of DenseNet, toward enhancing visual results, increasing security performance, and accelerating training. The ability to maintain high-quality visual characteristics and robust security even in resource-constrained environments, such as Internet of Things (IoT) contexts, demonstrates the practical benefits of this approach. The qualitative analysis of the visual results regarding the texture regions' localization and intensity exhibited augmented visual quality. Moreover, an improvement in the security attribute of 0.66% has also been demonstrated regarding average detection errors made by the SRM_EC Steganalyzer across all target payloads.**

***Keywords*—Content adaptive; distortion function; GAN; FC_DenseNet; steganography; steganalysis**

I. Introduction

Image steganography defines the art and science of hiding secret messages in digital images such that the intended recipient is most likely the only other entity aware of the secret [1] [2]. Numerous methods have been invented toward ensuring that such an intended secret is maintained. Over the past few years, image steganography's popularity has increased due to the vast amount of data transmitted over the Internet and social media platforms [3]. According to [2], adaptive image steganography is a promising new trend in the field of steganography that can engender greater assurance that such a secret will remain so. In content-adaptive image steganography, the embedding locations are modified adaptively based on the image's content, particularly its texture and smooth regions. To make the existence of a secret message undetectable, higher embedding probabilities are assigned to texture areas than are smooth areas. The most efficient embedding schemes employ content-adaptive steganography techniques that minimise the additive distortion function, as shown by Zhao et al. [4].

Minimizing embedding distortion simply means minimizing a well-designed additive distortion function, as defined here in Eq. (1) [5].

$$D(X,Y) = \sum_{i=1}^{W}\sum_{j=1}^{H}\{p_{i,j}\}\{x_{i,j} - y_{i,j}\} \tag{1}$$

Where, $D(X,Y)$ is the measure of the additive distortion caused by changing cover image X to Stego image Y. H and W are the height and width of the Stego and cover image, respectively. $p_{i,j}$ is the cost, or probability of changing pixel $x_{i,j}$ in cover image to $y_{i,j}$. P is a matrix representing the cost of changing or probability of changing pixel $x_{i,j}$ to $y_{i,j}$. The cost and the probability of change are inversely related.

Prior to discussing the methods of content adaptive steganography based on minimizing the distortion function, a review of the contrary field, specifically steganalysis, is necessary. It is worth noting that these two fields continuously impact one another. Steganalysis can be defined as the field in charge of detecting the existence of hidden information in an image. Initially, steganalysis was based on statistical methods [4]. However, with the advent of Machine Learning (ML) algorithms, steganalysis has evolved to employ ML algorithms thereby increasing detection accuracy by a focus on the feature extraction process. Fridrich et al. [6] proposed a Spatial Rich Model (SRM) utilizing 30 High-Pass Filters (HPF) to capture different relationships between neighboring pixels in different directions. SRM was enhanced to produce maxSRMd2 [7]

*Corresponding authors.

by defending against Selection Channel Aware steganalysis (SCA). SRM and maxSRMd2 extracted features are fed to a ML algorithm to perform the classification or equivalently the secret's detection. The Ensemble Classifier (EC) proposed by Kodovsky et al. [8] showed good performance back then.

In recent years, deep learning has become increasingly popular in image processing applications ushering in many innovative advancements. In particular, Convolutions Neural Network (CNN) is a powerful tool for extracting image features for both the spatial and frequency domains. In an effort to compete with the performance of features extraction handcrafted methods [6], [7], CNN-based Steganalyzers have been developed. In 2015, QIAN_Net [9], the first CNN-based Steganalyzer, was proposed. Their method uses a HPF in the preprocessing layer to strengthen steganographic noise. For feature extraction, five convolutions layers with a Gaussian activation function and average pooling are utilized for feature extraction. For the classification task, fully connected neurons were added to a softmax layer. QIAN_Net detection accuracy was inferior to hand-crafted methods [6], [7], which was the main motivation behind the proposal of Xu_Net [10] in 2016. In addition to the classification module, Xu_Net comprises various structural groups. Absolute, Batch Normalization (BN), and tanh are utilized in the initial groups to handle HPF output and improve statistical features. BN and Rectified Linear Unit (ReLU) are applied to the remaining groupings. Later, different ensemble strategies for the Xu_Net were investigated [11]. Instead of using a traditional HPF to determine steganographic noise, all 30 SRM HPF were utilized to initialize the kernels in the first convolution layer by Ye_Net [12]. Moreover, a Truncated Linear Unit (TLU) was proposed in an attempt to increase the Signal-to-Noise Ratio (SNR). Furthermore, incorporating knowledge of channel selection or the probability of changing each pixel provided for improved performance. Yedroudj_Net [13], an improved version of Ye_Net, was created by combining features from Xu_Net and Ye_Net. SR_Net [14] abandoned the idea of SRM preprocessing filters and initialized non-pooling convolutional layers randomly. SR_Net [14] achieved a state-of-the-art performance in 2018. ZHU_Net [15] enhanced their performance even further in 2019 by decreasing the kernel size to 3x3, using separable convolutions, and Spatial Pyramid Pooling (SPP). In 2021, GBRAS_Net [16] was proposed, which involves using filter banks to enhance steganographic noise in a preprocessing stage, depth wise and within separable convolutional layers, while skipping connections to the feature extraction stage. What makes GBRAS_Net different than all reviewed CNN-based Steganalyzers is the classification stage, which avoids overfitting by abandoning fully connected modules.

Despite the advancements in content-adaptive image steganography, current GAN-based methods exhibit several limitations that hinder their practical application. Specifically, these methods often neglect the semantic relationships between neighboring pixels, leading to suboptimal texture localization and security performance. Moreover, the prevalence of NaN-loss scenarios during training results in convergence issues, further degrading the quality of the generated stego images. Addressing these challenges is critical for enhancing the robustness and adaptability of steganographic techniques, particularly in resource-constrained environments, such as the Internet of Things (IoT). This research addresses the core problem of inefficient stego image generation in existing GAN-based steganographic models, stemming from their inability to effectively incorporate semantic information and mitigate training instability (e.g. NaN-loss scenarios). These challenges lead to a compromise in both the visual quality and security of the stego images, highlighting the need for an improved approach. The primary objective of this study is to propose an improved GAN-based framework, termed FT GAN, to overcome these limitations. By incorporating feature reuse through an FC DenseNet-based generator and introducing a bounded activation function to stabilize training, the proposed approach aims to enhance stego image quality, improve semantic segmentation, and ensure better security performance.

II. Related Work

The methods related to content adaptive image steganography are based on minimizing the additive distortion function by splitting the embedding process into two tasks. The first task objective is to generate a cost (or probability) matrix for each cover image using a distortion (or cost assignment) function that is well-designed. The goal of the second task is to produce Stego images using coding schemes such as Syndrome Trellis Codes (STC) [5], which take a cover image with its corresponding cost matrix and a secret message as inputs.

That being so, researchers developed various distortion or cost assignment functions, whose primary purpose is to achieve the first task and accurately assign the probability of change or cost of change by simply quantifying the effect of change, or $p_{i,j}$, for each pixel. Initially, the cost assignment functions were designed heuristically utilizing hand-crafted techniques, such as Highly Undetectable Stego (HUGO) [17], Wavelet Obtained Weights (WOW) [18], High pass, Low pass, and Low pass (HILL) [19], Spatial Universal Wavelet Retrieval Distortion (S_UNIWARD) [20], and Minimizing the Power of Optimal Detector (MiPOD) [21]. The previous handcrafted distortion functions provided a satisfactory level of security. Nevertheless, their primary insufficiency was that the detectability factor was not considered when designing the cost function. According to Pevny et al. [17], the cost of embedding is directly related to its detectability. However, simulating this correlation was practically impossible back then.

With the development of GAN [22], it became possible to simulate the distortion and detectability relationship. Tang et al. [23] were the first to automatically design a distortion function. Automatic Steganographic Distortion Learning using a Generative Adversarial Network (ASDL_GAN) was proposed by Tang et. al. [23] in 2017. Their approach included three parts: Generator G, a Ternary Embedding Simulator (TES), and Discriminator D. Their generator was comprised of 25 structural groups, with each group containing a convolution layer, BN layer, and ReLU activation function, while a shortcut was utilized to identify the feature map of the stack layers. The process takes as input the cover image and the target capacity for which embedding probabilities are to be produced. The TES is used to simulate ternary data embedding since the vanishing gradient problem prohibits the conventional staircase function from being used directly. The TES takes as inputs the probability map matrix produced by the generator and a matrix of floating-point integers representing the secret message, and

returns a modification map, which produces a Stego image when added to the cover image that has a better metrics.

Tang et. al. [23]'s TES was a mini-network requiring a long pre-training time. Therefore, Yang et al. [24] improved on this aspect by proposing a double tanh_simulator in 2018. Moreover, motivated by the fact that ASDL_GAN security performance was inferior to hand-crafted distortion functions and the U_Net [25] capabilities in pixel-wise segmentation they proposed a new generator based on U_Net. Moreover, to resist SCA based steganalysis, they incorporate SCA into the discriminator adopting Xu_Net's architecture similar to ASDL_GAN. Yang et. al. [26] modified their earlier 2019 work, thereby investigating the influence of high pass filters in the discriminator's preprocessing layer to consequently propose UT_6HPF_GAN.

Despite the fact that the UT_SCA_GAN [24] and UT_6HPF _GAN [26] performed similar to or better than conventional methods, according to Tang et al. [27], the vanishing gradient problem still exists after several iterations. This problem follows from using the sigmoid/tanh activation function as embedding simulators which prevent the full exploitation of the architecture's potential. Thus, Steganographic Pixel-Wise Actions and Rewards with RL (SPAR_RL) architecture was proposed in 2019 by Tang et al. [27]. In this approach, a policy network attempted to learn an embedding policy by decomposing the embedding into pixel-wise actions to maximize rewards. A sampling process was designed to simulate the embedding actions, and the gradients of data embedding were allocated to the reward function. Tang et al. [27] were able to alleviate the vanishing gradient problem in SPAR_RL. However, they ignored the semantic information between neighboring pixels as can be observed in the policy network of SPAR_RL. Additionally, existing architectures overlook the NAN-loss scenarios that prevent proper convergence and, thus degrade the Stego image visual quality. These issues also relate to the poor adaptability of existing works from ignoring feature-reuse, useful for pixel-wise segmentation, as well as texture localization in the Stego images. Therefore, we consider these issues herein, and have redesigned the GAN's generator for improved image steganography.

III. An Improved GAN Architecture for Improved Image Steg

Briefly, our work has improved the GAN architecture to address the main problem of SPAR_RL, while preserving the generator's key goal of creating high-quality, and secure Stego images. To this end, the GAN's architecture was improved by utilizing the semantic segmentation neural networks for generators other than U_Net. Inspired by the superior semantic segmentation capabilities of FC_DenseNet [28], feature reuse, implicit deep supervision, and the vanishing gradient problem alleviation of DenseNet [29], the model's convergence and image quality is significantly improved. The proposed model was evaluated against both the FC_DenseNet [28] and U_Net in [25] an encoder-decoder architectures (or CNN-based) image steganography.

The hypothesis asserts that the FC_DenseNet internal connections will allow feature reuse to be carried by the features map to the subsequent layers. Further, this has been shown to enhance coarse semantic feature extraction and texture localization in images. As a result, the security level will be enhanced.

To this end, the main contribution of this paper is threefold:

1) Developing FT_GAN by incorporating the FC _DenseNet feature reuse into the GAN's generator, increases the quality of the generated image, as well as the average Stego image security.
2) Improving the architecture using a bounded-activation-function, prevents NAN-loss and enhances the model convergence and image visual quality.
3) The performance of the improved model proposed here was evaluated against existing architectures in terms of detection error, and image visual quality for the model's security and imperceptibility judged to produced comparatively better results.

IV. Materials and Methods

A. Dataset and Software Platforms

The following data set has been utilized during experimentation. All images have been scaled to 256x256 in an effort to accelerate the training and preserve resources.The Google Colab pro+ platform was utilized to perform these experiments.

1) BOSSBase v1.01: used for the earlier contest of breaking steganographic system, containing 10000 images of size 512x512, as well as used to test GAN,
2) BOWS#2: used for 10000 images of size 512x512 (first used for a contest to break watermarking systems).

Each of the previous datasets has been permuted randomly at a ratio of 8:2. This ratio gives how many images were used for GAN training, GAN testing, which includes SRM training and testing.

1) GAN training uses 16,000 images, 8000 of which come from BOSSBase and another 8000 from BOWS2;
2) GAN testing consists of 4,000 images, which are divided into 50% SRM training and 50% SRM testing.

B. Overall Architecture of FT_GAN

The overall architecture of the FT_GAN is shown below, refer to Fig. 1. The architecture is composed of a generator, a ternary embedding simulator, and a discriminator. As described, the architecture is same as [23], [24], [26], specifically [26] with the only difference being in the generator design.

The process begins by feeding the cover image to the generator to produce its corresponding probability map, which is then passed to the ternary embedding simulator along with the input stream representing the secret message to generate the modification map. The modification map is added to the cover image to obtain the Stego image. The pair is then input to the Xu_Net [10] discriminator after passing through a six SRM high pass filter, to perform classification. Finally, the loss made by the generator and discriminator is computed to update GAN's weights.

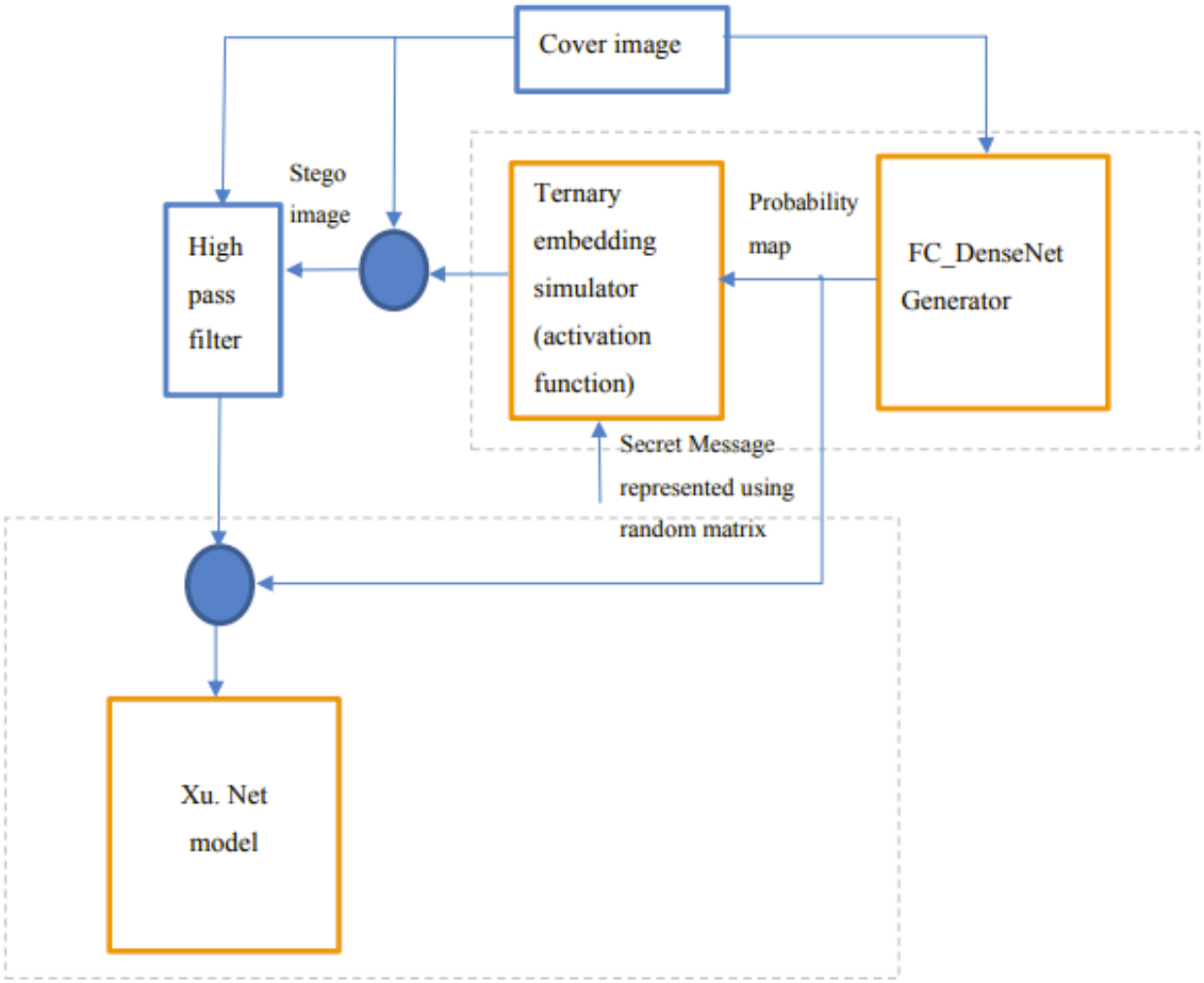

Fig. 1. The overall architecture.

1) FC_DenseNet_102 _D7-based Generator, and its loss function: In previous work, the U_Net architecture has been utilized for the generator [24], [26], this design enabled results superior to hand-crafted methods. An attempt to increase the performance even further by investigating FC_DenseNet architecture in the generator is made here. The main motivation is the comparison held by Duan et. al., [30] in 2021 when they compared the performance of FC_DenseNet [28] and U_Net [25] in CNN-based image steganography.

The FC_DenseNet_102_D7-based generator architecture is summarized in Fig. 2. This architecture attempts to maintain U_Net depth while incorporating the feature reuse capability of FC_DenseNet. A default growth rate of 16 is maintained, and the number of layers per dense block in every level is adjusted to match the width of U_Net layer in the same level. In this way, an investigation of feature reuse capability to improve performance and accelerate learning can be accomplished. To avoid NAN-loss, GAN convergence failure mode and improve the visual results quality, a bounded activation function is proposed here, and a pre-processing layer of a small kernel width is utilized.

The generator's loss function consists of two parts: the adversarial and entropy parts. The adversarial loss, which seeks to increase security performance, is the negative of the discriminator loss. The primary purpose of the entropy part is to ensure the embedding payload is met by the resultant probability maps. Refer to Eq. (2), where α and β are scaling factors and set to 1 and $\frac{1}{10^7}$, respectively. I_D is calculated using binary cross entropy, Refer to Eq. (3), Where y_i is the softmax, or discriminator output, and y_i' is truth label Stego/cover. Alternatively, I_C is computed using Eq. (4), where H, W, and Q are height, width, and target payload, respectively. Capacity is calculated with help from a generator produced probability map. Refer to Eq. (5) [4].

$$I_G = -\alpha I_D + \beta I_C^2, \tag{2}$$

$$I_D = \sum_{i=1}^{2} y_i' \log y_i, \tag{3}$$

Pre- processing layer		**Feature Map (WXHXC)**
input		256x256x1
Convolution: Kernel size=3x3 Number of kernels=1		256x256x1
Model name	**Architecture**	**Feature Map**
FC_DenseNet_102 _D7	DB (1L) + TD	128x128x17
	DB (2L) + TD	64x64x49
	DB (4L) + TD	32x32x113
	DB (8L) + TD	16x16x241
	DB (8L) + TD	8x8x369
	DB (8L) + TD	4x4x497
	DB (8L) + TD	2x2x625
	DB(8L)	2x2x753
	TU+DB(8L)	4x4x881
	TU+DB(8L)	8x8x753
	TU+DB(8L)	16x16x625
	TU+DB(8L)	32x32x497
	TU+DB(4L)	64x64x305
	TU+DB(2L)	128x128x145
	TU+DB(1L)	256x256x65
	1x1Conv	256x256x1
Post Processing layers	**Bounded-Activation-Function.** ReLU $\left(Sigmoid * 0.5 - \left(2^{-125}\right)\right) + \left(2^{-125}\right)$	256x256x1
	Output	256x256x1

Fig. 2. The architecture of FC_DenseNet_102_D7-based generator.

$$I_C = Capacity \times H \times W \times Q, \tag{4}$$

$$Capacity = \sum_{i=1}^{H}\sum_{j=1}^{W} -p_{i,j}\log_2\frac{p_{i,j}}{2} - (1-p_{i,j})\log_2(1-p_{i,j}), \tag{5}$$

2) Ternary Embedding Simulator (TES): The TES attempts to simulate the ternary embedding operation, refer to Eq. (6). The ternary embedding operation (TEO) takes as input the pixel's probability map $p_{i,j}$, and a floating-point value $n_{i,j}$ obtained from uniform distribution of (0,1), representing a secret message. The TEO output's a modification value $m_{i,j}$, which is then added to cover pixel's value x_i,j to produce Stego pixel value $y_{i,j}$.

$$m_{i,j} = \begin{cases} -1, \text{ if } n_{i,j} < \frac{p_{i,j}}{2} \\ 1, \text{ if } n_{i,j} < 1 - \frac{p_{i,j}}{2} \\ 0, \text{otherwise} \end{cases}, \tag{6}$$

The fact that the stair case function defined by Eq. (6) or the TEO are not differentiable and that they do not preserve the gradient loss during back-propagation, is the main motivation behind utilizing the double-Tanh TES proposed by Yang et al. [24] during experiment. Refer to Eq. (7), where λ is a controlling factor equal to 60 [26].

$$m_{i,j} = -0.5Tanh(\lambda(p_{i,j} - 2n_{i,j})) + 0.5Tanh(\lambda(p_{i,j} - 2(1-n_{i,j}))) \tag{7}$$

3) Discriminator and its loss functions: The Xu_Net architecture is adopted for the discriminator [10]. Xu Net comprises a preprocessing module, a convolution module, and a classification module. The preprocessing module made use of six SRM HPF [26]. The convolution module is made up of structural groupings of the convolution, activation, and pooling operations. Absolute, Batch normalization (BN), and tanh are used in the early groups to manage high pass filter output and enhance statistical characteristics. The remaining groupings utilize BN and ReLU. In the classification module, a fully connected layer and softmax activation are utilized. The discriminator weights are updated with the help of the binary cross-entropy loss function defined at Eq. (3).

C. GAN Training, and Hyper-parameters

To conserve resources and provide a fair comparison between UT_GAN and FT_GAN, a GAN training dataset is fed in batches of eight during the experiment. The Adam optimizer with a 0.001 learning rate is applied to update the generator's weight. The Discriminator optimizer is a stochastic gradient descent with a fixed momentum of 0.9, and initial learning rate of 0.001. Thus, the process was scheduled to decrease by 10% every 5,000 iterations. All generator weights were initialized with random values drawn from a normal distribution with zero mean and 0.02 standard deviation. Similarly, the Discriminator convolution kernel weights are initialized randomly from a normal distribution, but with standard deviation of 0.01. However, the fully-connected (FC) layers parameters were initialized using a "Xavier" initialization.

This previous architecture characterization and hyper-parameters were used to train the GAN for a 0.4bpp target payload. Subsequently, this model was fine-tuned for other target payloads using curriculum learning (CL).

D. Evaluation

1) Visual Evaluation: The FC_DenseNet_102_D7-based generator has been evaluated qualitatively based on the clarity and location of the generated probability map and modification map. Also, the convergence speed is an important consideration, which is the rate at which these clear, localized visual results, start to show up. Fig. 4, and 5 show visual results for the 0.4bpp Target payload, and all other Target payloads respectively.

2) Security Evaluation: The FC_DenseNet_102_D7-based generator security performance was evaluated with the help of the SRM_EC [6], [8]. GAN testing data has been split in half. The first half has been used to train SRM_EC. The second half has been used to test SRM_EC. [6], [8] Eq. (8) was used to compute the average detection error over ten trials of Ensemble classifier training and testing. The final results are shown in Table I. Here, P_{FA} is the number of false alarms processed by the by SRM_EC in cover images while P_{MD} is the number of missed detections from Stego images.

$$P_E = \frac{1}{2}(P_{FA} + P_{MD}) \tag{8}$$

V. Results

A. Visual Results

The figures below describe the main visual results. Fig. 4 compares the visual results of FT_GAN and UT_GAN during training for a 0.4bpp target payload at various epochs. Similarly, Fig. 5 compares them, but for different target payloads.

B. Security Results

Table I summarizes the average detection error made by SRM_EC for all trained payloads. The last column in the table shows the average P_E across all target payloads. Refer to Fig. 3.

VI. Discussion

A. Visual Results Discussion

The visual results discussion is conveyed in terms of a qualitative analysis for the probability map and modification map summarized in Fig. 4 and 5.

The probability map is superior if the majority of white regions, representing regions with a high likelihood of embedding, are located in the texture region of the image. Clearly, these observations, seen from the figures show that the FC_DenseNet_102_D7-based generator probability maps' more intense white compared to the U_Net-based generator probability maps' white, indicate that the texture areas (of the prior) were assigned the highest probability value, which is 0.5. Recall that the probability value range is (0,0.5).

TABLE I. AVERAGE DETECTION ERROR MADE BY SRM_EC FOR ALL TARGET PAYLOADS USING GAN TESTING DATA

Steganography	**0.1bpp**	**0.2bpp**	**0.3bpp**	**0.4bpp**	**0.5bpp**	**0.6bpp**	**0.7bpp**	**Average Among All Payloads**
UT_GAN	0.1493	**0.1550**	0.1508	**0.1536**	**0.1562**	**0.1588**	0.1362	0.1514
FT_GAN	**0.1508**	0.1470	**0.1539**	0.1498	0.1556	0.1567	**0.1532**	**0.1524**[1]

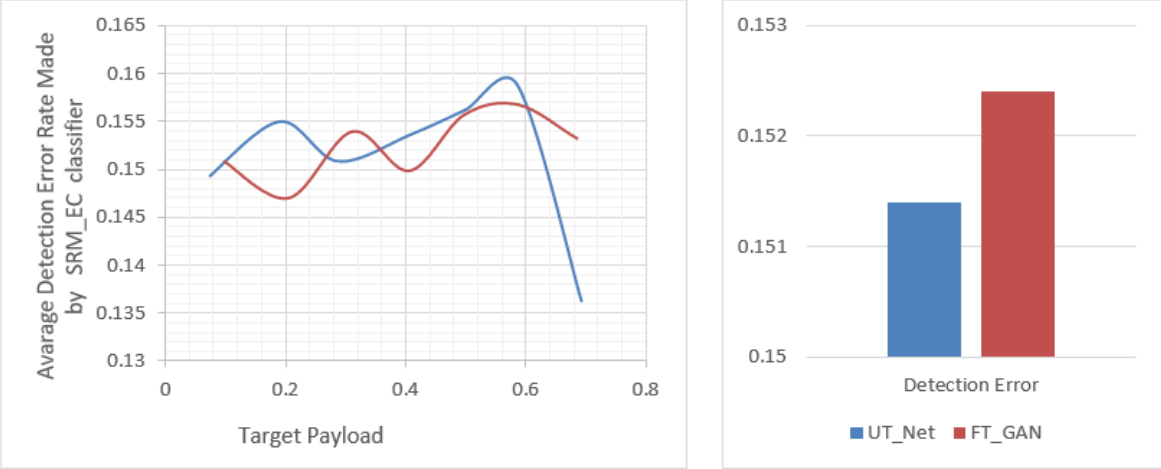

Fig. 3. Missed detection made by SRM_EC across each target payload, and the average across all of them.

From Fig. 4, we can conclude that the FC_DenseNet_102_D7-based generator began localizing texture regions faster than U_Net-based generator beginning at epoch 20. This indicates the benefit of feature reuse. The modification map is also better if it displays all black "-1" and white "+1" in the texture areas. Smooth areas represented by gray "0" remain unchangeable. Recall that the modification map is obtained using Eq. (7). On the basis of this accounting, a better modification map corresponds to a better probability map's localization.

B. Security Results and Discussion

Recall that our experimental dataset is not similar to those dataset(s) used by Yang et al.'s [26] since the ZSUBase dataset is not publicly available. Therefore, the comparison is held with the architecture proposed by Yang et al. [26] when trained using our aforementioned experiment dataset. According to the Table I, and Fig. 3, the security performance of the two architectures varies, making it difficult if not impossible for the user to determine which architecture outperforms. This is true as seen from the outcome of several variables, including the amount of training iterations and fine-tuning. However, these parameters are extremely important in GAN training due to the min-max game it plays. This game produces fluctuations in a variety of metrics, including security results. For instance, the value at one epoch may be quite high, but significantly fall in a subsequent epoch. This phenomenon leads us to understand that a precise decision criterion is needed. Thus, a CNN-based steganalysis is required to precisely determine the necessary number of fine-tuned epochs and iterations to optimize the best and most accurate final results. During these experiments, this precise decision criterion was unavailable owing to a lack of dataset size and resources. Aside from the number of fine-tuned epochs and iterations, this fluctuation was also caused by other SRM_EC factors, such as the number of base learners and the d_sub.

The experimental decision criteria was mostly visual, in addition to the loss made in meeting the target capacity, refer to Eq. (4). Therefore, the comparison is based on the last column of the table, or the "other" average classification error

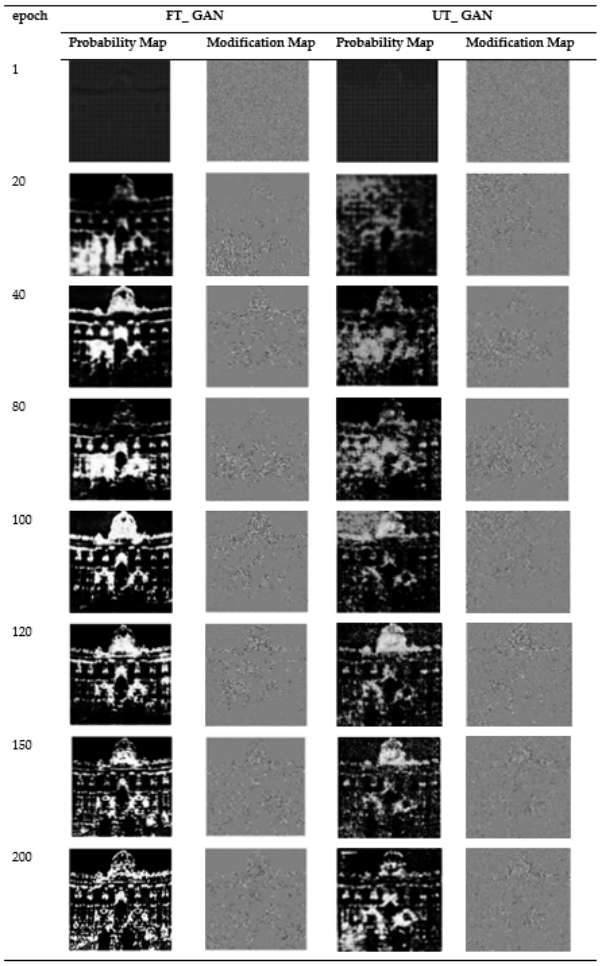

Fig. 4. Visual results for target payloads of 0.4bpp in terms of probability map and modification map.

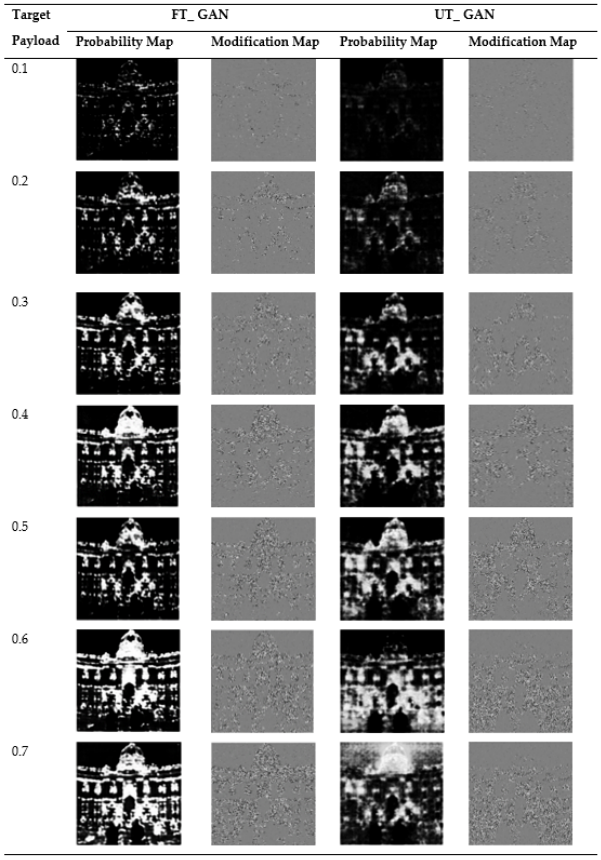

Fig. 5. Visual results for all target payloads in terms of probability map and modification map.

made across all trained payloads. Refer to Fig. 4. Overall, the FC_DenseNet_102_D7-based generator (or FT_GAN superior U_Net-based) generator improved the detection error by 0.66%. This minor average enhancement is a result of the FC_DenseNet_102_D7 architecture, and more specifically the reuse of features. This is supported by the visual result at both Fig. 4 and 5.

When comparing the results described in Eq. (1) with those of earlier studies [26], it is clear that the detection error rate reported by this experiment is lower, even for their proposed design, namely U_Net. This demonstrates conclusively that it was not the outcome of the FC_DenseNet_102_D7-based generator design, but rather the dataset employed (i.e. a limited dataset). According to Karras et al. [31], limited data is one of the challenges of GAN training. The fundamental problem with limited datasets is that the discriminator rapidly overfits to the training examples.

Consider that the discriminator's function is to classify its inputs as either cover or Stego. But, due to overfitting, it rejects as Stego all inputs other than the initial training dataset (cover images)! As a result, the generator receives minimal input to assist in enhancing the quality of its subsequent output, and thus, rendering the training process worthless. Karras et al. [31] explain why longer training did not eliminate the +1 and -1 points in the smooth regions throughout the experiment. Refer to Fig. 4, and 5. The experimental results clearly demonstrate the superiority of the proposed FT GAN over existing models in terms of both visual quality and security. For instance, the faster convergence speed observed with the FT GAN underscores the benefit of feature reuse and deep supervision provided by the FC DenseNet architecture. This capability allows the generator to focus embedding efforts on textured regions more effectively, as evidenced by the intense white areas on the probability maps (see Fig. 4). This improved localization directly enhances the stego images' imperceptibility, reducing detectability by steganalyzers. Compared to UT GAN and other GAN-based steganography models proposed by Yang et al. [24, 26], the FT GAN demonstrates a more consistent performance across all payload sizes. Specifically, the average detection error for FT GAN (0.1524) is marginally but consistently better than UT GAN (0.1514), as shown in Table I. This improvement highlights the practical advantage of incorporating FC DenseNet's feature reuse capabilities in the generator architecture, which is absent in UT GAN. While previous models relied on U-Net or similar architectures, the FT GAN's design mitigates vanishing gradient problems and achieves more robust outputs. The results also emphasize the importance of addressing training instability, a common limitation in earlier works like ASDL GAN [23] and UT 6HPF

GAN [26]. By introducing a bounded activation function, the FT GAN significantly reduces NaN-loss scenarios, leading to smoother training convergence and higher-quality outputs. In contrast, earlier models often struggled with overfitting or unstable training, particularly when using small datasets.

VII. Conclusions

We have presented our FT_GAN for content-adaptive image steganography, developed by incorporating feature reuse into the GAN generator. FT_GAN has been evaluated based on both visual and security results using SRM_EC Steganalyzers. The main outcome of this work is a clear improvement in the visual results of the FC_DenseNet_102_D7-based generator over the U_Net-based generator using BOSSBase and BOWS2 datasets, as well as an average security improvement of 0.66% compared to all other target payloads. As a future recommendation, we highly endorse the development of a universal generator that satisfies both the spatial and JPEG domains, since the one proposed here works only in the spatial domain. By leveraging FC DenseNet and a bounded activation function, our approach demonstrated improved visual quality, faster convergence, and enhanced security, with lower detection error across all payloads. Future work will focus on extending FT GAN to support the JPEG domain, exploring larger datasets for scalability, and integrating advanced learning techniques to further optimize embedding strategies and expand its practical applications in secure communication systems.

References

[1] N. Subramanian, O. Elharrouss, S. Al-Maadeed, and A. Bouridane, "Image steganography: A review of the recent advances," *IEEE access*, vol. 9, pp. 23 409–23 423, 2021.

[2] I. J. Kadhim, P. Premaratne, P. J. Vial, and B. Halloran, "Comprehensive survey of image steganography: Techniques, evaluations, and trends in future research," *Neurocomputing*, vol. 335, pp. 299–326, 2019.

[3] I. Hussain, J. Zeng, X. Qin, and S. Tan, "A survey on deep convolutional neural networks for image steganography and steganalysis," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 14, no. 3, pp. 1228–1248, 2020.

[4] J. Zhao and S. Wang, "A stable gan for image steganography with multi-order feature fusion," *Neural Computing and Applications*, pp. 1–16, 2022.

[5] T. Filler, J. Judas, and J. Fridrich, "Minimizing additive distortion in steganography using syndrome-trellis codes," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 920–935, 2011.

[6] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.

[7] T. Denemark, V. Sedighi, V. Holub, R. Cogranne, and J. Fridrich, "Selection-channel-aware rich model for steganalysis of digital images," in *2014 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2014, pp. 48–53.

[8] J. Kodovsky, J. Fridrich, and V. Holub, "Ensemble classifiers for steganalysis of digital media," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 432–444, 2011.

[9] Y. Qian, J. Dong, W. Wang, and T. Tan, "Deep learning for steganalysis via convolutional neural networks," in *Media Watermarking, Security, and Forensics 2015*, vol. 9409. SPIE, 2015, pp. 171–180.

[10] G. Xu, H.-Z. Wu, and Y.-Q. Shi, "Structural design of convolutional neural networks for steganalysis," *IEEE Signal Processing Letters*, vol. 23, no. 5, pp. 708–712, 2016.

[11] G. Xu, H.-Z. Wu, and Y. Q. Shi, "Ensemble of cnns for steganalysis: An empirical study," in *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, 2016, pp. 103–107.

[12] J. Ye, J. Ni, and Y. Yi, "Deep learning hierarchical representations for image steganalysis," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2545–2557, 2017.

[13] M. Yedroudj, F. Comby, and M. Chaumont, "Yedroudj-net: An efficient cnn for spatial steganalysis," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 2092–2096.

[14] M. Boroumand, M. Chen, and J. Fridrich, "Deep residual network for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 5, pp. 1181–1193, 2018.

[15] R. Zhang, F. Zhu, J. Liu, and G. Liu, "Depth-wise separable convolutions and multi-level pooling for an efficient spatial cnn-based steganalysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1138–1150, 2019.

[16] T.-S. Reinel, A.-A. H. Brayan, B.-O. M. Alejandro, M.-R. Alejandro, A.-G. Daniel, A.-G. J. Alejandro, B.-J. A. Buenaventura, O.-A. Simon, I. Gustavo, and R.-P. Raul, "Gbras-net: a convolutional neural network architecture for spatial image steganalysis," *IEEE Access*, vol. 9, pp. 14 340–14 350, 2021.

[17] T. Pevnỳ, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," in *International workshop on information hiding*. Springer, 2010, pp. 161–177.

[18] V. Holub and J. Fridrich, "Designing steganographic distortion using directional filters," in *2012 IEEE International workshop on information forensics and security (WIFS)*. IEEE, 2012, pp. 234–239.

[19] B. Li, M. Wang, J. Huang, and X. Li, "A new cost function for spatial image steganography," in *2014 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2014, pp. 4206–4210.

[20] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP Journal on Information Security*, vol. 2014, no. 1, pp. 1–13, 2014.

[21] V. Sedighi, R. Cogranne, and J. Fridrich, "Content-adaptive steganography by minimizing statistical detectability," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 2, pp. 221–234, 2015.

[22] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets (advances in neural information processing systems)(pp. 2672–2680)," *Red Hook, NY Curran*, 2014.

[23] W. Tang, S. Tan, B. Li, and J. Huang, "Automatic steganographic distortion learning using a generative adversarial network," *IEEE Signal Processing Letters*, vol. 24, no. 10, pp. 1547–1551, 2017.

[24] J. Yang, K. Liu, X. Kang, E. K. Wong, and Y.-Q. Shi, "Spatial image steganography based on generative adversarial network," *arXiv preprint arXiv:1804.07939*, 2018.

[25] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[26] J. Yang, D. Ruan, J. Huang, X. Kang, and Y.-Q. Shi, "An embedding cost learning framework using gan," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 839–851, 2019.

[27] W. Tang, B. Li, M. Barni, J. Li, and J. Huang, "An automatic cost learning framework for image steganography using deep reinforcement learning," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 952–967, 2020.

[28] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 11–19.

[29] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[30] X. Duan, N. Liu, M. Gou, W. Wang, and C. Qin, "Steganocnn: image steganography with generalization ability based on convolutional neural network," *Entropy*, vol. 22, no. 10, p. 1140, 2020.

[31] S. M. Thomas, J. G. Lefevre, G. Baxter, and N. A. Hamilton, "Towards highly expressive machine learning models of non-melanoma skin cancer," *arXiv preprint arXiv:2207.05749*, 2022.