Image Encryption and Compression Based on Reversed Diffusion Model

Yilin Guo*, Jianhui Chang†, Yuhuai Zhang†, Jian Zhang*, Siwei Ma†
ylguo@stu.pku.edu.cn, {jhchang, yhzhangvcl, zhangjian.sz, swma}@pku.edu.cn
**School of Electronic and Computer Engineering, Peking University*, Shenzhen, China
†*School of Computer Science, Peking University*, Beijing, China

***Abstract*—Nowadays, as critical conduits of communication, the information security of images and videos is particularly important. The existing encryption techniques usually transform images into high-frequency content that resembles noise, presenting significant challenges in achieving efficient compression. This paper presents an innovative collaborative approach that integrates image encryption and compression using a reversed diffusion model. This method, by reversing the typical process of diffusion models, adeptly changes encrypted high-frequency content into a domain that is more amenable to compression. Leveraging the reversible nature of the Denoising Diffusion Implicit Models (DDIM), our framework ensures the high-fidelity restoration of information. Our experimental findings demonstrate that this approach not only effectively encrypts images but also compresses the encrypted high-frequency noise content, outperforming Video Versatile Coding (VVC) in compression performance.**

***Index Terms*—Image encryption, encrypted image compression, and diffusion models.**

I. Introduction

In today's digital age, the omnipresence of images and videos increases the risk of privacy violations and cyber threats, making data protection and privacy crucial. At present, image encryption [1–3], steganography [4, 5] and watermarking [6] are receiving attention for maintaining image data security. Image steganography and watermarking are techniques for hiding information within another medium, such as embedding a secret image or watermark into another picture, without obvious alteration. The hidden results are still readable and significantly impacted by the compression process. In contrast, image encryption [1, 2, 7] transforms images into a secure and unreadable format using specialized techniques like chaotic or displacement transformations to protect the information. Nevertheless, the task of compressing encrypted images remains a significant challenge, primarily because these encrypted images inherently possess high-frequency noise, which is not suitable for normal compression algorithms and may degrade the quality of the compressed image.

Currently, traditional image compression technologies have played a fundamental role in visual communication and image processing, bringing up a series of crafted image compression codecs, such as JPEG [8], WebP [9], and VVC reference software VTM [10]. Beyond traditional codecs and standards, deep learning based image compression have gained prominence [11–14]. In the realm of digital image compression, traditional and learning-based coding methods exhibit high efficiency for conventional content, but face limitations with encrypted images. Traditional hybrid coding operates by transforming images into the frequency domain, capitalizing on the human eye's heightened sensitivity to low-frequency details. This approach typically involves discarding high-frequency information to reduce bit rates. Encrypted images, primarily high-frequency, are ill-suited for traditional compression methods that target lower frequencies. On the other hand, learning-based coding methods transform images into a domain characterized by low-correlation features, aiding in compression. Nonetheless, a challenge arises with encrypted content, which typically consists of largely random noise with low correlation, leading to minimal compressibility in end-to-end coding systems. Consequently, encrypted images with high-frequency noise present a substantial challenge in coding, and an effective solution for their compression remains elusive.

The development of diffusion models [15, 16] presents promising opportunities for advancing both encryption and compression technologies. These models incrementally introduce and reverse noise in data, enabling high-quality generative applications in image synthesis and restoration [17]. The forward process is stochastic, adding Gaussian noise, while recent developments reveal a deterministic reverse process, effective for embedding information within this noise domain, thus facilitating image encryption [17]. Inspired by the dual information extraction philosophy [17, 18], we propose to encrypt an image into a spatial domain characterized by Gaussian noise, coupled with a high-level compact semantic feature vector, leveraging the reverse diffusion process for encoding. To address the challenge of compressing encrypted images, predominantly characterized by high-frequency noise, we propose to invert the conventional diffusion process by using the generation phase for encoding purposes while adopting the noise-adding phase as its generative mechanism. Consequently, this method efficiently diminishes random noise and converts encrypted images into a more compressible domain, while ensuring security through the separate extraction of semantic information.

In this work, we introduce a novel collaborative framework for image encryption and compression based on the reversed diffusion model. This framework encompasses two diffusion models: the outer diffusion, adhering to a conventional diffusion model structure, encrypts the image and concurrently extracts semantic information; and the inner diffusion, strate-

2024 Picture Coding Symposium (PCS) | 979-8-3503-5848-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/PCS60826.2024.10566414

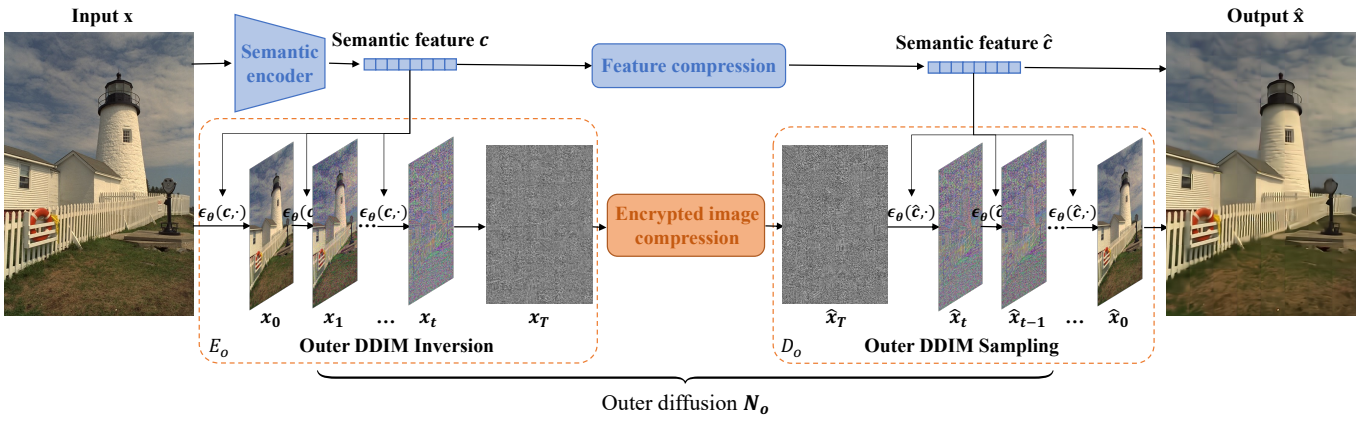

Fig. 1. **Architecture of the proposed encryption and compression framework**. It illustrates the workflow from input to output, consisting of outer diffusion, feature encoder and compression, inner diffusion and compression. The input $\mathbf{x}$ undergoes encryption through outer diffusion inversion. After encryption, the encrypted image and its associated semantic features are compressed through deep learning-based entropy coding modules.

gically designed for denoising the encrypted image before encoding and adding noise for decoding. Subsequently, we employ learning-based entropy coding skills to further compress the encrypted image and its associated semantic features. On the decoder side, we achieve image reconstruction by inverting the encoding process of both the inner and outer diffusion models. Experimental evidence indicates that this approach not only ensures robust image encryption but also achieves more effective compression of encrypted information, maintaining a lower bitrate compared to conventional compression codecs.

II. Proposed Method

We aim to introduce/remove noise from images by utilizing conditional DDIM, facilitating the complete image encryption, compression, and decryption processes. The pipeline of the proposed framework is presented in Fig. 1, consisting of the outer diffusion N_o, feature compression module, encrypted image compression module and a semantic encoder Enc_ϕ. The encrypted image compression module includes the inner diffusion N_i and a lightweight learning-based entropy codec C_i, while the features compression module has another lightweight learning-based feature codec C_s. In the encryption stage, the input image $\mathbf{x}$ first undergoes iterative noise addition through the outer DDIM inversion process E_o of N_o, resulting in the encrypted image x_T and the semantic feature c. Subsequently, x_T is compressed using the encrypted image compression module, while c is compressed through the feature compression module. In the decryption stage, the reconstructed $\hat{x_T}$ and $\hat{c}$ are re-entered into DDIM sampling process D_o of N_o to reconstruct image $\hat{\mathbf{x}}$.

A. Outer Diffusion

The outer diffusion model N_o is the main network for image encryption and decryption. The purpose of image encryption and decryption is to reconstruct the image after encryption into an unreadable image, which can be summarized as a type of image inverse problem. Numerous approaches have been developed for image inverse problems, among which diffusion models are gaining prominence, pushing the limits of solving these complex issues.

DDIM [16] proves to achieve the comparable generation capability as the original diffusion model [15] by rewriting the sampling formula while accelerating the sampling time and providing capabilities for image inverse problem solving and image compression. The sampling formula is defined as follows:

$$x_{t-1} = \sqrt{\alpha_{t-1}}x_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\epsilon_\theta(x_t,t) + \sigma_t\epsilon \quad (1)$$

where the definition of the parameters can be found in [16], $\epsilon_\theta(x_t,t)$ denotes the predicted noise, ϵ denotes the random noise and the coefficients before the noise term can all be set to zero.

Therefore, we find that if the coefficient of the last random noise term is set to $\mathbf{0}$, the added noise for each sample is deterministic and non-random under fixed network output. During the inversion process, we estimate the noise at the same time step based on the network and perform noise addition operations. When the total step number is large enough and the time interval is small, the "inversion-sampling" process in the DDIM can be considered reversible. Based on the reversibility of the DDIM, we reverse the inversion and sampling processes of the diffusion model.

Inspired by DiffAE [17], we separate the sampling and inversion processes of DDIM for performing image encryption and decryption respectively. Firstly the semantic encoder extracts the corresponding semantic features $c = \mathrm{Enc}_\phi(\mathbf{x})$, as the condition of DDIM. Then the outer DDIM inversion, or E_o, converts the original image into a highly abstract, unreadable and high-noise encrypted image x_T under the guidance of c. The inversion formula is defined as follows:

$$\mathbf{f}_\theta(\mathbf{x}_t,t,c) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \sqrt{1-\alpha_t}\epsilon_\theta(\mathbf{x}_t,t,c)\right) \quad (2)$$

$$\mathbf{x}_{t+1} = \sqrt{\alpha_{t+1}}\mathbf{f}_\theta(\mathbf{x}_t,t,c) + \sqrt{1-\alpha_{t+1}}\epsilon_\theta(\mathbf{x}_t,t,c) \quad (3)$$

where the setting of α, σ is the same as in [16] and $\mathbf{x}_0$ is the original image $\mathbf{x}$, which means the starting point of the DDIM inversion process.

When decrypting, $\hat{x}_T$ and $\hat{c}$ are fed into the DDIM iterative sampling network D_o to achieve high-fidelity image decryp-

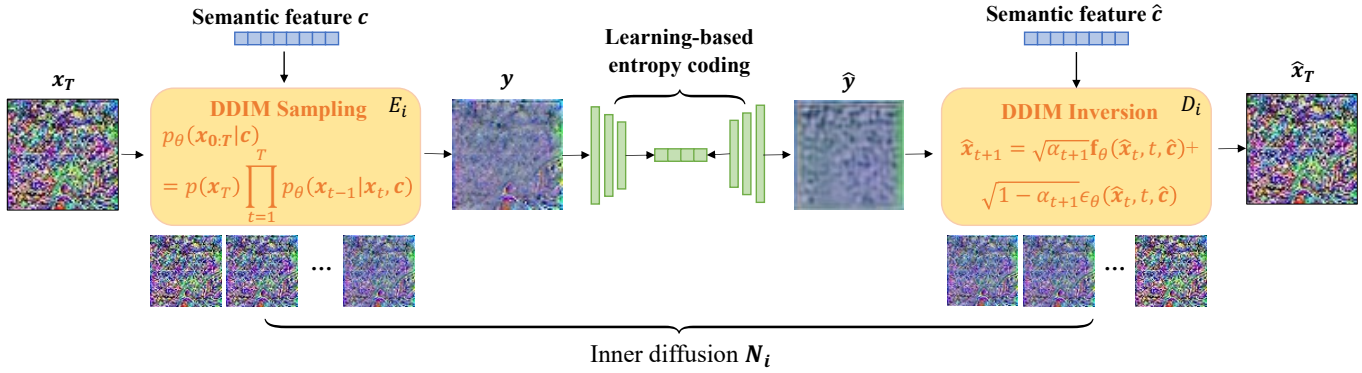

Fig. 2. **Detailed process of encrypted image compression**. The proposed method includes a reversed diffusion model for reducing encrypted image random noise and a lightweight learning-based entropy coding module. In particular, we employ DDIM sampling at the encoder side, which diminishes random noise and converts encrypted images into a more compressible domain. On the decoder side, DDIM inversion is applied to restore the encrypted image.

tion. The sampling formula is similar to Eq. (1), specifically as follows:

$$\hat{\mathbf{x}}_{t-1} = \sqrt{\alpha_{t-1}}\hat{\mathbf{x}}_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\epsilon_\theta\left(\hat{\mathbf{x}}_t, t, \hat{c}\right) \quad (4)$$

where $\hat{\mathbf{x}}_0$ is estimation of the reconstructed image $\hat{\mathbf{x}}$, which is progressively corrected in each iteration.

The image is divided into a 64×64 sub-block sequence for encryption, and the resulting encrypted sub-blocks are randomly arranged to form the final encrypted image, while the real sub-block number is hidden within each block.

B. Inner Diffusion

The framework of the inner diffusion is presented in Fig. 2. We designed an iterative denoising and compression method for the encrypted images of high-noise based on the reversed diffusion.

According to Fig. 2, under the semantic feature condition c, we iteratively sample the high-noise encrypted image x_T to eliminate high-frequency information and obtain the denoised spatial representation y. The iteration formula of the inner sampling process E_i closely resembles Eq. (5). To distinguish it from outer diffusion, we denote the result of each iteration by x_t:

$$x_{t-1} = \sqrt{\alpha_{t-1}}x_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\epsilon_\theta\left(x_t, t, c\right) \quad (5)$$

where we set the iteration to start from x_T and continue downwards until $t = 0$. Ultimately, this process results in y.

Then C_i further performs transformation, quantization and entropy coding on y. During the decoding process, the reconstructed $\hat{y}$ is gradually added with approximate noise of the sampling step to achieve the reconstruction of $\hat{x}_T$, following the same formula as outlined in Eq. (2) and Eq. (3). Through this approximately reversible process, the elimination and restoration of high-frequency information are achieved.

C. Feature Compression

In the proposed framework, both encrypted images and their associated semantic information play crucial roles in image decryption. After the encrypted image is converted into a more compressible spatial feature by E_i, we employ a learning-based entropy coding module C_i to further compress them into the bitstream. C_i consists of multiple residual blocks and the space-channel context entropy model SCCTX [19] for entropy coding. Similarly, the semantic information is further compressed using the learning-based compression method C_s. C_s utilizes 1×1 convolution to decrease the dimension of c and incorporates hyper-prior entropy model [20] to compress features.

D. Loss of framework

Overall, the proposed framework contains two diffusion models and two CNN encoders. The sampling of the diffusion model is separated from the inversion process, but inseparable during pre-training. Therefore, our loss mainly consists of three parts: diffusion loss, Rate-Distortion (R-D) loss, and fine-tuning loss.

Diffusion loss. Diffusion loss is the fundamental loss for training outer and inner diffusion models. Similar to DiffAE [17], the initial training of N_o and N_i is accomplished by optimizing the L_{simple}[15] loss function with respect to θ and ϕ.

$$L_{\text{simple}} = \sum_{t=1}^{T}\mathbb{E}_{\mathbf{x}_0,\epsilon_t}\left[\left\|\epsilon_\theta\left(\mathbf{x}_t, t, c\right) - \epsilon_t\right\|_2^2\right] \quad (6)$$

where $\epsilon_t \in \mathbb{R}^{3\times h\times w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1-\alpha_t}\epsilon_t$, $1<t<T$, and T is set to 1,000.

N_o, N_i are pre-trained through diffusion loss and then fine-tuned with the reconstruction loss.

Rate-Distortion loss. Rate-Distortion (R-D) loss refers to the loss of the training image and feature codecs, which involves a trade-off between bitrate and distortion rate. Since y represents the crucial spatial information of the original image, the image codec C_i utilizes MSE and SSIM for distortion measurement. On the other hand, c represents the semantic information in the form of features and should maintain high fidelity. Therefore, the feature codec C_s employs the L1 loss to measure distortion.

$$L_{\text{codec}} = \mathcal{R}_j + \lambda_j * \mathcal{D}_j, j \in \{C_i, C_s\} \quad (7)$$

$$\mathcal{D}_{C_i} = MSE(y, \hat{y}) + SSIM(y, \hat{y}) \quad (8)$$

$$\mathcal{D}_{C_s} = L_1(c, \hat{c}) \quad (9)$$

where $\mathcal{R}_j$, $\mathcal{D}_j$ and λ_j in Eq. (7) express estimate rate, distortion and balance parameters of C_i and C_s.

Fine-tuning loss. The entire framework involves the reconstruction of the input $\mathbf{x}$, the encrypted spatial information x_T, the denoised spatial information y, and the semantic information c. Due to the inherent distortion introduced by codecs, directly using raw parameters that are not exposed to lossy y and c for decoding will result in errors. Therefore, it is necessary to fine-tune the inversion process of N_i and the sampling process of N_o to ensure accurate decoding.

$$L_{\mathrm{f}} = \lambda_i * MSE(x_T, \hat{x_T}) + \lambda_o * MSE(\mathbf{x}, \hat{\mathbf{x}}) \qquad (10)$$

where λ_i, λ_o are the weighting parameters to balance each component. We empirically set $\lambda_i = 0.1$ and $\lambda_o = 10$.

Note that $L_{\text{simple}}, L_{\text{codec}}$ and L_{f} are used independently during different training phases rather than being combined. Detailed training procedures are provided in Section III.

III. Experiments

A. Experimental Settings

Implementation details. We trained the networks using Adam Optimizer with β_1 = 0.9 and β_2 = 0.999 on NVIDIA GeForce RTX 3090 GPUs. Our training process involved several stages. Initially, we employed L_{simple} to train the outer diffusion N_o for the encryption task. Then we froze the parameters of N_o and trained the inner diffusion N_i using L_{simple}. Subsequently, we separately trained two compression modules, C_i and C_s, using L_{codec}. Finally, we integrated the entire pipeline and fine-tuned the decoding parameters D_i and D_o using L_{f} to optimize overall performance. Moreover, our diffusion model operates at a patch resolution of 64×64. Prior to further processing, the input image is segmented into blocks and ultimately reconstructed to its original resolution.

Datasets. In each training epoch, we randomly crop 64x64 pixel blocks extracted from the Open Image v4 validation dataset [21], which encompasses a diverse collection of 41,620 images featuring a range of resolutions and scene compositions. And we use the Kodak dataset [22] for testing.

B. Performance of Image Encryption

According to Fig. 3, in the case of lossless compression, the reconstructed image achieves a PSNR of 41.2057, indicating a high quality of decryption. Besides, the average color value analysis can easily detect slight differences in color contributions that form an image. The original image has an uneven average of the three colors, while in an encrypted image, the average value of RGB values is uniform, preventing the attacker from interpreting any information from it. As a result, the developed algorithm is secure in terms of image encryption and decryption.

Moreover, the effectiveness of the framework can be quantitatively demonstrated by calculating the information entropy. For an ideal random image, its information entropy is equal to 8. According to Fig. 3, the information entropy of the encrypted images is 7.43, closely resembling that of a pure noise image. The formula of information entropy is as follows:

$$H(X) = -\sum_{i=0}^{L-1} p\left(x_i\right) \log_2 p\left(x_i\right) \qquad (11)$$

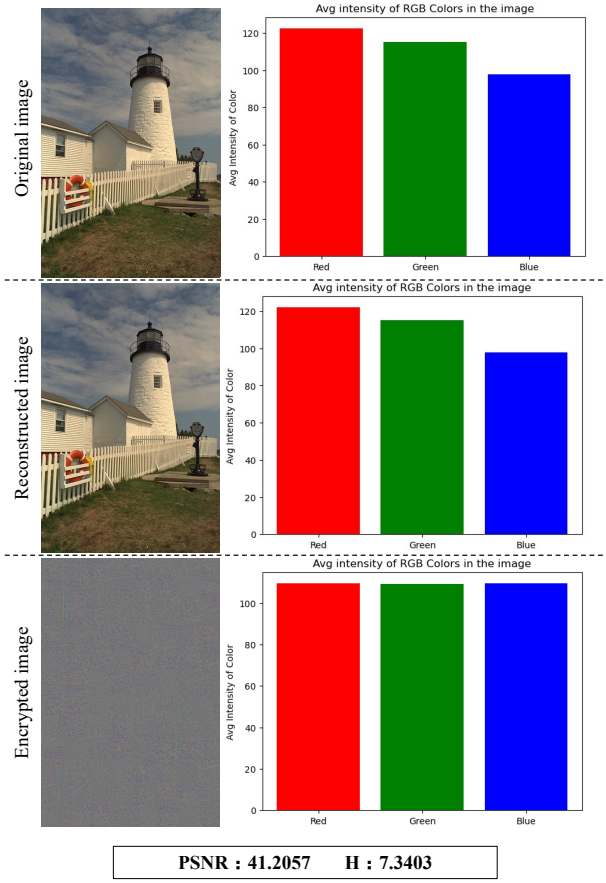

Fig. 3. **Average intensity of RGB channels in original, reconstructed, and enciphered images.** The average values across each channel reveal minor variations in the image's color contributions. As the image grows more chaotic and disordered, the distribution differences between the channels lessen.

where L is the number of color values (L = 256), $p(x_i)$ is the probability that a color value appears.

C. Performance of Image Compression

To verify the effectiveness of the proposed image compression scheme, we compare it with the standard codec JPEG [8], WebP [9] and VTM (Intra-mode) [10]. We evaluate compression performance in terms of bits per pixel (BPP), SSIM, PSNR and LPIPS. Among these, SSIM and PSNR represent a relatively objective evaluation metric, while LPIPS proves to be highly related to subjective evaluation. According to Table I, our method outperforms JPEG, WebP and VTM (Intra-mode) on the Kodak dataset, achieving higher subjective and objective decryption performance with lower bit rates.

IV. Conclusion

In conclusion, this study addresses the critical issue that the encrypted content of traditional encryption methods is difficult to compress due to high-frequency noise. By innovatively applying a reversed diffusion model, we effectively reformat encrypted high-frequency content into a more compressible structure. Utilizing DDIMs, our framework ensures accurate data restoration. Experimental results demonstrate that our

TABLE I
PERFORMANCE OF THE PROPOSED METHOD, JPEG, WEBP AND VVC ON THE KODAK DATASET. HIGHER PSNR AND SSIM SCORES AND LOWER LPIPS SCORES INDICATE LOWER DISTORTION.

Method	BPP ↓	SSIM↑	PSNR↑	LPIPS↓
Ours	**0.5298**	**0.9301**	20.4634	**0.6043**
VVC	0.6287	0.9285	19.7082	0.6104
WebP	0.6087	0.9280	**20.5963**	0.6173
JPEG	0.6086	0.9218	19.5376	0.6234
Ours	**2.0719**	**0.9555**	**23.1854**	**0.3840**
VVC	2.1722	0.9513	22.8820	0.3868
WebP	2.2508	0.9534	23.1050	0.4555
JPEG	2.6506	0.9452	22.6790	0.3913

approach not only encrypts images effectively but also compresses the encrypted content more efficiently than VVC, enhancing both performance and efficiency.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China (NSFC) No. 62031013, and in part by the Xplore Prize, which are gratefully acknowledged.

REFERENCES

[1] F. Hu, C. Pu, H. Gao, M. Tang, and L. Li, "Image compression and encryption scheme based on deep learning," *Natsional'nyi Hirnychyi Universytet. Naukovyi Visnyk*, no. 6, pp. 142–148, 2016.

[2] X. Duan, J. Liu, and E. Zhang, "Efficient image encryption and compression based on a VAE generative model," *Journal of Real-Time Image Processing*, vol. 16, pp. 765–773, 2019.

[3] H. R. Latha and A. Ramaprasath, "HWCD: A hybrid approach for image compression using wavelet, encryption using confusion, and decryption using diffusion scheme," *Journal of Intelligent Systems*, vol. 32, no. 1, p. 20229056, 2023.

[4] Y. Xu, C. Mou, Y. Hu, J. Xie, and J. Zhang, "Robust invertible image steganography," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7875–7884.

[5] J. Yu, X. Zhang, Y. Xu, and J. Zhang, "CRoSS: Diffusion model makes controllable, robust and secure image steganography," in *Proceedings of the Advances in Neural Information Processing Systems*, 2023.

[6] M. Begum and M. S. Uddin, "Digital image watermarking techniques: a review," *Information*, vol. 11, no. 2, p. 110, 2020.

[7] Z. Bao, R. Xue, and Y. Jin, "Image scrambling adversarial autoencoder based on the asymmetric encryption," *Multimedia Tools and Applications*, vol. 80, no. 18, pp. 28 265–28 301, 2021.

[8] W. B. Pennebaker and J. L. Mitchell, *JPEG: Still image data compression standard*. Springer Science & Business Media, 1992.

[9] Z. Si and K. Shen, "Research on the WebP image format," in *Advanced graphic communications, packaging technology and materials*. Springer, 2016, pp. 271–277.

[10] B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.-R. Ohm, "Overview of the versatile video coding (VVC) standard and its applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3736–3764, 2021.

[11] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," in *International Conference on Learning Representations*, 2017.

[12] S. Ma, X. Zhang, C. Jia, Z. Zhao, S. Wang, and S. Wang, "Image and video compression with neural networks: A review," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 6, pp. 1683–1698, 2019.

[13] J. Ballé, P. A. Chou, D. Minnen, S. Singh, N. Johnston, E. Agustsson, S. J. Hwang, and G. Toderici, "Nonlinear transform coding," *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 2, pp. 339–353, 2020.

[14] S. Ma, J. Gao, R. Wang, J. Chang, Q. Mao, Z. Huang, and C. Jia, "Overview of intelligent video coding: from model-based to learning-based approaches," *Visual Intelligence*, vol. 1, no. 1, pp. 1–15, 2023.

[15] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[16] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations*, 2021.

[17] K. Preechakul, N. Chatthee, S. Wizadwongsa, and S. Suwajanakorn, "Diffusion autoencoders: Toward a meaningful and decodable representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 619–10 629.

[18] J. Chang, Z. Zhao, C. Jia, S. Wang, L. Yang, Q. Mao, J. Zhang, and S. Ma, "Conceptual compression via deep structure and texture synthesis," *IEEE Transactions on Image Processing*, vol. 31, pp. 2809–2823, 2022.

[19] D. He, Z. Yang, W. Peng, R. Ma, H. Qin, and Y. Wang, "ELIC: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5718–5727.

[20] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *International Conference on Learning Representations*, 2018.

[21] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov *et al.*, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *International Journal of Computer Vision*, vol. 128, no. 7, pp. 1956–1981, 2020.

[22] R. Franzen, "Kodak lossless true color image suite. 1999," *source: http://r0k. us/graphics/kodak*.