Image Data Hiding in Neural Compressed Latent Representations

Chen-Hsiu Huang and Ja-Ling Wu

Dept. of Computer Science and Information Engineering

National Taiwan University, Taipei, Taiwan

{chenhsiu48,wjl}@cmlab.csie.ntu.edu.tw

Abstract—We propose an end-to-end learned image data hiding framework that embeds and extracts secrets in the latent representations of a generic neural compressor. By leveraging a perceptual loss function in conjunction with our proposed message encoder and decoder, our approach simultaneously achieves high image quality and high bit accuracy. Compared to existing techniques, our framework offers superior image secrecy and competitive watermarking robustness in the compressed domain while accelerating the embedding speed by over 50 times. These results demonstrate the potential of combining data hiding techniques and neural compression and offer new insights into developing neural compression techniques and their applications.

Index Terms—Image steganography, watermarking, neural compression, end-to-end learned data hiding

I. INTRODUCTION

Image steganography or watermarking hides secrets in a cover image to form a container image for communication or proof of ownership. Steganography focuses on the container image's secrecy and message capacity, while watermarking techniques must be robust to various attacks. Traditional methods worked on hiding and extracting secrets in either the spatial domain [1], [2] or the frequency domain [3], [4]. Modern techniques [5]–[8] use deep neural networks (DNNs) and adversarial training to end-to-end learn an encoder/decoder pair that embeds and extracts the secrets with robustness against noise attacks. These data hiding techniques are highly relevant to image compression codecs and DNN-transformed latent representations.

Neural compression [9], the end-to-end learned image compression method [10]–[13], has been actively developed in recent years and has proven to outperform traditional expert-designed image codecs. Although the model complexity and performance issues remain the challenges for neural compression to be widely adopted, international standards such as JPEG AI [14] and MPEG VCM (Video Coding for Machines) [15] have initiated to bridge data compression and computer vision together for both human and machine vision. Choi et al. [16] have proposed scalable image coding frameworks based on well-developed neural compressors to achieve up to 80% bitrate savings on machine vision tasks.

Data compression itself may not justify the necessity to replace handcrafted image/video codecs with learned approaches. However, the intersection of data compression and multimedia applications may offer us a different perspective. In this work, we propose an end-to-end learned image data hiding framework that embeds and extracts secrets in the latent representations of a neural compressor.

II. RELATED WORKS

A. Learned Image Compression

DNNs have opened new opportunities to rebuild data compression as an end-to-end learning process. While several approaches existed in the literature, Balle et al. [10] proposed a highly successful image codec that out-performed JPEG and JPEG 2000 in PSNR and SSIM metrics. Minnen et al. [11] then used a joint autoregressive and hierarchical prior model to achieve even higher coding efficiency than the HEVC [17] codec. More recently, Cheng et al. [12] developed techniques that achieve comparable performance to the latest coding standard VVC [18]. There are now several excellent survey and introduction papers [9], [19], [20] summarizing this wave of end-to-end compression advances.

B. DNN-based Steganography and Watermarking

Deep steganography methods like DeepStega [5] and UDH [6] defined a new task to hide one or more images into a cover image with a DNN. Unlike traditional methods that require a perfect restoration of secret messages, Deep steganography methods minimize the distortion between the retrieved and the original secret images. Thus, the message is securely delivered because the authentic and recovered secret images are visually indistinguishable. Lu et al. [21] recently advanced deep steganography with a higher capacity of up to three or more secret images.

Zhu et al. [7] proposed the HiDDeN model that embeds the raw bits and extracts the secret message with a low bit error rate using a DNN. With the generative model adversarial trained against the noise layers, the HiDDeN model can achieve the purposes of steganography and digital watermarking with the same network architecture. Perhaps inspired by HiDDeN, many researchers have proposed similar adversarial network methods for steganography, such as StegaStamp [8], Hinet [22], and watermarking [23]–[25].
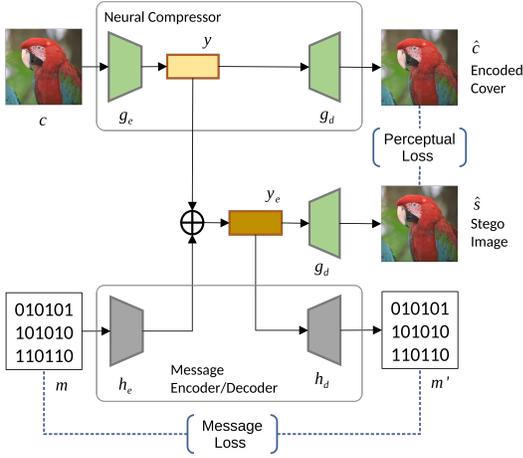
Fig. 1. The architecture of the proposed neural data hiding framework.

III. PROPOSED METHOD

We show our system architecture in Fig. 1 and describe our techniques as follows.

A. Problem Formulation

a) Steganography: A message $m \in \{0,1\}^n$ is hidden in a cover image c. A neural encoder/decoder pair g_e and g_d are used to obtain the compressed latent vector $y = g_e(c)$ and the encoded cover image $\hat{c} = g_d(y)$. A message encoder h_e is trained to transform the message to the same dimension as y. The embedded latent vector y_e is obtained as follows:

$$y_e = y \oplus h_e(m) \tag{1}$$

where $\oplus$ is element-wise addition. The embedded latent vector y_e is then entropy-coded and transmitted. The stego image $\hat{s} = g_d(y_e)$ can be decoded by anyone with access to the publicly available g_e and g_d. However, only the specified receiver with a trained message decoder h_d can extract the secret message by:

$$m' = h_d(y_e) \tag{2}$$

b) Watermarking: The noised stego image $\dot{s} = u(\hat{s})$ is derived using an attacker u to simulate noise attacks. The final noised stego image $\hat{\dot{s}}$ needs to be re-compressed using the same neural encoder/decoder pair. The noised embedded latent vector $\dot{y}_e = g_e(\dot{s})$ is then used to extract secrets with $\dot{m}' = h_d(\dot{y}_e)$.

B. Network Architecture

We design our message encoder h_e and decoder h_d to match neural codecs [10] and [11]'s latent representations. We detail the encoder/decoder structures in Table I. Different neural codecs may have varying latent space dimensions, but the general rules are: 1) Message encoder: Use a linear layer and a CBNL (convolution, batch normalization, and leaky ReLU) layer to expand the $|m|$-bit message to match the compressed latents' dimensions. 2) Message decoder: Use Conv/ReLU layers with downsampling and then flatten and use linear regression to obtain the $|m|$-bit message.

TABLE I
NETWORK ARCHITECTURE

Message Encoder h_e						
Layers	kernel	stride	padding	in	out	channels
Linear				$|m|$	64	
CBNL	3	1	1	8×8	8×8	320
Message Decoder h_d						
Layers	kernel	stride	padding	in	out	channels
Conv/ReLU	3	2	1	8×8	4×4	320
Conv/ReLU	3	2	1	4×4	2×2	320
Conv/ReLU	3	2	1	2×2	1×1	320
Flatten						
Linear				320	512	
Linear				512	$|m|$	

C. Loss Functions

We define our loss function as a combination of perceptual loss $\mathcal{L}_P$ and message loss $\mathcal{L}_M$:

$$\mathcal{L} = \mathcal{L}_P + \alpha \mathcal{L}_M, \tag{3}$$

where α is a hyper-parameter used to control the relative weight of the two losses.

The perceptual loss is measured by the DNN-based perceptual loss LPIPS [26]:

$$\mathcal{L}_P = \text{LPIPS}(\hat{c}, \hat{s}). \tag{4}$$

We avoid using MSE (mean square error) to minimize image distortion because we observed that the LPIPS metric significantly impacts our fixed neural codec more than a self-trained image encoder/decoder.

For measuring the decoded message error, we use binary cross-entropy as the loss function:

$$\mathcal{L}_M = \text{BCE}(m, m') + \beta \text{BCE}(m, \dot{m}'), \tag{5}$$

where $\dot{m}'$ is the decoded message from the noised latent $\dot{y}_e$. The loss function is weighted by hyper-parameters $\alpha = 1.5$ and $\beta = 1.0$ in our experiments.

D. Noise Attacks

For the watermarking scenario, we defined four types of common noise attacks as Cropout, Dropout, Gaussian noise, and JPEG compression. We randomly generate the noise parameters during training.

IV. EXPERIMENTAL RESULTS

We implemented our works on neural codecs hyper [10] and mbt [11] from CompressAI. We denote our data hiding methods as "Ours-hyper" and "Ours-mbt," respectively. To ensure high visual quality after encoding, we set the highest coding quality as 8 in both codecs.

For training, we randomly selected 12,000 and 1,200 images from the COCO dataset [28] as the training and validation set, respectively. We resized the cover images to 128×128 and randomly embedded 32-bit binary messages during training.

We trained our model using the PyTorch built-in Adam optimizer with a learning rate of 0.001 and a batch size of 32. We trained our model for 160 epochs. We compared our model with HiDDeN[1], DeepStega, UDH[2], and StegaStamp[3]. Unlike other DNN-based methods that calculate distortion between the cover c and stego image s, we measured the distortion between $\hat{c}$ and $\hat{s}$, as described in Section III-A.

A. Steganography Secrecy

Quantitatively, we present image quality metrics in PSNR, SSIM, MAE (mean absolute error), and bit error rate, as shown in Table II. While it is well-known that DNN-based methods cannot achieve zero bit error rates, there are established techniques, such as BCH codes [29] and learning-based channel noise modeling [30], to mitigate this issue.

Our evaluation accounts for the effect of quantization on latent vectors. As both the hyper and mbt neural codecs use unit scalar quantization, the modified latent coefficients can still withstand the quantization operation. Table II indicates that our proposed methods have less perceptual distortion than others. The superior stego image quality of the hyper codec stems from its lower coding efficiency than the mbt codec, allowing more room for data hiding in the latent space. On the other hand, the mbt codec has more densely compressed latents, which results in more significant quality degradation from the quantization operation.

TABLE II
QUALITY METRICS VS. BIT ERROR RATE COMPARISON

Kodak [31]				
Method	**PSNR↑**	**SSIM↑**	**MAE↓**	**Error**
Ours-hyper [10]	[1]43.44	[1]0.9942	[1]1.02	0.00000
Ours-mbt [11]	[2]40.48	[2]0.9881	[2]1.65	0.00000
HiDDeN [7]	39.61	0.9813	1.91	0.00000
DeepStega[a] [5]	36.51	0.9374	2.81	0.01564
UDH[a] [6]	37.88	0.9184	2.63	0.02131
StegaStamp[b] [8]	31.60	0.9430	4.54	0.00500
DIV2K [32]				
Method	**PSNR↑**	**SSIM↑**	**MAE↓**	**Error**
Ours-hyper	[1]41.67	[1]0.9945	[1]1.36	0.00000
Ours-mbt	[2]38.42	[2]0.9847	[2]2.25	[c]0.00094
HiDDeN	37.59	0.9733	2.44	0.00125
DeepStega	34.72	0.9283	3.58	0.01839
UDH	38.35	0.9414	2.51	0.02475
StegaStamp	30.50	0.9451	5.38	0.00890
CelebA [33]				
Method	**PSNR↑**	**SSIM↑**	**MAE↓**	**Error**
Ours-hyper	[1]46.08	[1]0.9962	[1]0.75	0.00000
Ours-mbt	[2]40.56	0.9768	2.25	0.00000
HiDDeN	39.28	[2]0.9806	[2]1.85	0.00000
DeepStega	38.27	0.9410	2.34	0.01822
UDH	38.27	0.9147	2.54	0.01502
StegaStamp	35.40	0.9586	2.67	0.00460

[a]Use the change of MSB to report bit error
[b]Image size 400×400, embed 100 bits
[c]Bit errors due to quantization

[1]https://github.com/ando-khachatryan/HiDDeN

[2]https://github.com/ChaoningZhang/Universal-Deep-Hiding
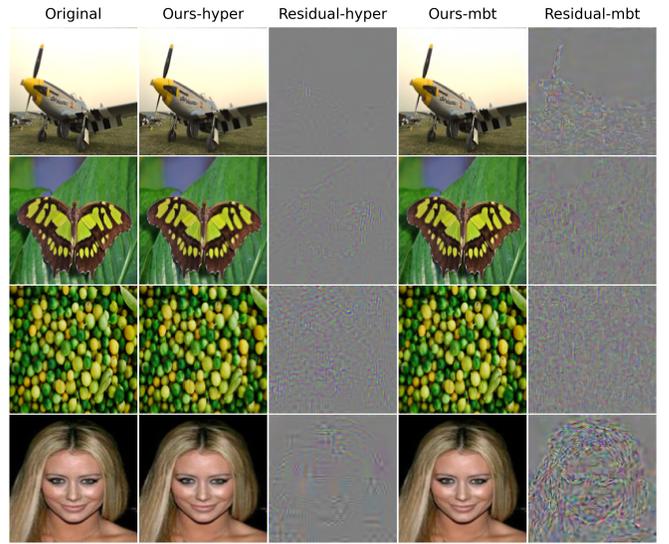
[3]https://github.com/tancik/StegaStamp

Fig. 2. Comparison of stego image quality and residual. Please zoom in to observe the modified pixel locations.
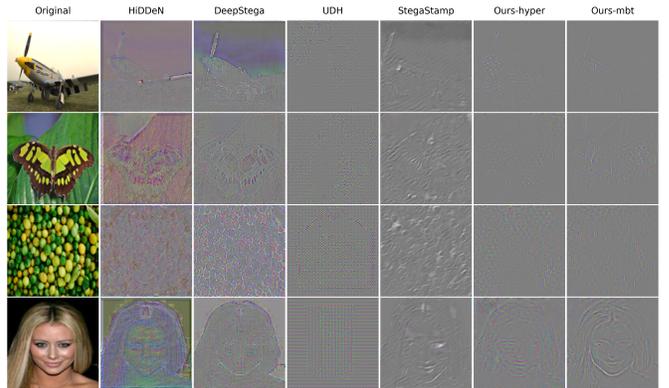

Fig. 3. The stego image residual comparison of different data hiding methods. Please zoom in to observe the modified pixel locations.

Qualitatively, we present the cover images and the resulting stego images in Fig. 2 and compare our methods' stego image residual with other DNN-based methods in Fig. 3.

a) Pixel modification: Modern DNN-based data hiding methods add perturbations to the low-level feature space to extract messages from the spatial domain with robustness. As a result, these methods modify all the low-level pixels of the cover image, as shown in Fig. 3. Our neural data hiding method learns to modify the compressed latents, so the pixel modifications are placed in high-level image features, as shown in Fig. 2. Overall, our proposed method can generalize well on different neural codecs and has a less perceptual impact.

Although in our scenario, the receiver does not have access to the encoded cover image $\hat{c}$, we list the LPIPS [26] metrics between c and $\hat{s}$ in Table III and compare them with those of other methods. Our LPIPS metrics remain close to those of the HiDDeN method and are superior the other methods. The LPIPS is a learned perceptual metric based on a pre-trained
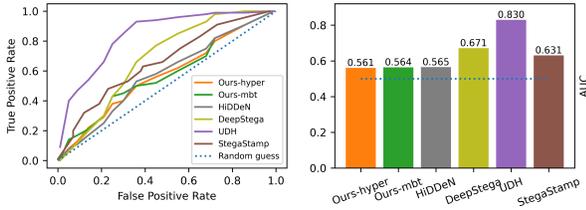
Fig. 4. The ROC curve of StegExpose classifier.

Fig. 5. Watermark robustness against selected noise attacks, evaluated on DIV2K.

DNN, which can be thought of as how effectively the stego image can be used as a proxy for the original cover image. Therefore, we believe the generated stego image $\hat{s}$ has not lost its general utility.

TABLE III
LPIPS COMPARISON OF STEGO IMAGES ON DIV2K

Method	LPIPS$(\hat{c}, \hat{s})$	LPIPS$(c, \hat{s})$	LPIPS(c, s)
Ours-hyper	0.00064	0.00392	-
Ours-mbt	0.00485	0.00599	-
HiDDeN	-	-	0.00375
DeepStega	-	-	0.07718
UDH	-	-	0.04261
StegaStamp	-	-	0.08039

b) Steganalysis: We measured the ability of our model to resist steganalysis using publicly available steganalysis tools, including traditional statistical methods [34] and new DL-based approaches [35] [36]. To assess our model's anti-steganalysis ability on the DIV2K dataset, we used the steganalysis tool StegExpose [34].

We varied the detection thresholds as input to StegExpose and plotted the ROC (receiver operating characteristic) curve. We then calculated the AUC (area under the curve) to indicate the classification effectiveness. Ideally, the AUC should be close to 0.5, indicating that the classifier performs no better than random guessing. Figure 4 shows the ROC curve and the AUC of the compared methods. Our proposed Ours-hyper method achieved slightly better secrecy than HiDDeN, with an AUC of 0.561.

B. Watermark Robustness

We evaluated the robustness of our method against trained noise attacks on the DIV2K dataset, as shown in Fig. 5. We varied the attack strength by increasing the noise parameter, which degrades image quality along the horizontal axis. In the last chart of Fig. 5, we observe that the overall PSNR is lower than the numbers we reported in Table II, which is the trade-off we make for achieving robustness.

Our neural data hiding method performs equivalently to HiDDeN in the cropout attack, as the cropped-out pixels do not provide any information to the message extractor. However, for attacks such as dropout, Gaussian noise, and JPEG compression, the proposed method demonstrates superior robustness to spatial domain extraction methods like HiDDeN and StegaStamp. We believe this is due to modifying of the compressed latent space, which retains more information in the high-level features without being impacted.
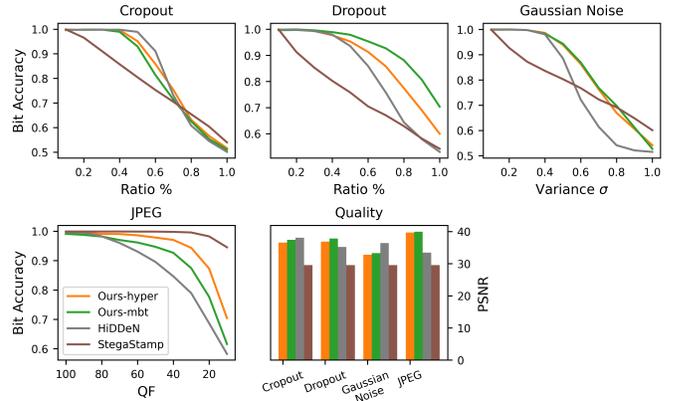
C. Embedding Performance

We report the message embedding/extraction time in Table IV, measured on an Intel i7-9700K workstation with an Nvidia GTX 3090 GPU. As expected, our method embeds messages 50 times faster than other spatial domain approaches.

TABLE IV
EMBED/EXTRACT TIME ON DIV2K, IN SECONDS.

Method	Embed	Extract
Ours-hyper	0.00021	0.00051
Ours-mbt	0.00020	0.00049
HiDDeN	0.01040	0.00074
DeepStega	0.01108	0.00071
UDH	0.01091	0.00070
StegaStamp	0.05893	0.03412

TABLE V
MESSAGE EMBEDDING SIZE OVERHEAD, IN PERCENTAGE.

Method	Kodak	DIV2K	CelebA
Ours-hyper	15.62%	11.95%	25.61%
Ours-mbt	20.51%	14.38%	31.18%
HiDDeN[a]	4.36%	5.03%	9.71%

[a]In compressed PNG

D. Neural Codecs

We point out some key considerations to be taken into account when building applications on neural compressed latent representations:

First, the continuous latents will be quantized before entropy coding. Although in our case, the modified latent coefficients can still withstand unit quantization being used, it is worth further investigating how different quantization operations can affect the message embedding process.

Second, the impact on coding efficiency. The neural compressor learns a compact representation and a near-optimal probability estimation end-to-end. Adding perturbations to latent coefficients inevitably breaks the compressor's learned optimal coding strategy. The overheads shown in Table V range from 11% to 31% depending on the dataset and the underlying codecs used.

We did not jointly optimize the image codec with the message encoder/decoder because we aimed to prove the concept of neural data hiding in a standard neural compressor. Third, we observed that a more coding efficient neural compressor leads to more quality degradation from quantization and more size overhead after embedding.

E. Perceptual Loss Study

The success of message extraction is highly dependent on the design of the message encoder/decoder when jointly trained with a standard codec that minimizes image distortion. Experimental results show that the widely used MSE loss performs poorly. Interestingly, we observed that the LPIPS metric significantly impacts our fixed neural codec more than a self-trained image encoder/decoder. This observation could be due to the high compactness of our neural codec, which is more strongly connected to certain perceptual features in the latent space.

V. CONCLUSION

In this work, we proposed a novel end-to-end framework for image data hiding that embeds secrets in the latent representations of a neural compressor. Our approach is generic and can be used with different neural compressors. We demonstrated its superior image secrecy and competitive watermarking robustness while significantly accelerating the embedding speed. Future researches could focus on designing more efficient neural compressors that reserve space for data hiding in their latent vectors.

REFERENCES

[1] T. Pevný, 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.

[2] 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.

[3] J. Fridrich, T. Pevný, and J. Kodovský, "Statistically undetectable jpeg steganography: dead ends challenges, and opportunities," in *Proceedings of the 9th workshop on Multimedia & security*, 2007, pp. 3–14.

[4] N. Bi, Q. Sun, D. Huang, Z. Yang, and J. Huang, "Robust image watermarking based on multiband wavelets and empirical mode decomposition," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 1956–1966, 2007.

[5] S. Baluja, "Hiding images in plain sight: Deep steganography," *Advances in Neural Information Processing Systems*, vol. 30, pp. 2069–2079, 2017.

[6] C. Zhang, P. Benz, A. Karjauv, G. Sun, and I. S. Kweon, "Udh: Universal deep hiding for steganography, watermarking, and light field messaging," *Advances in Neural Information Processing Systems*, vol. 33, pp. 10 223–10 234, 2020.

[7] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "Hidden: Hiding data with deep networks," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 657–672.

[8] M. Tancik, B. Mildenhall, and R. Ng, "Stegastamp: Invisible hyperlinks in physical photographs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2117–2126.

[9] Y. Yang, S. Mandt, and L. Theis, "An introduction to neural data compression," *arXiv preprint arXiv:2202.06533*, 2022.

[10] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," *arXiv preprint arXiv:1802.01436*, 2018.

[11] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," *Advances in Neural Information Processing Systems*, vol. 31, pp. 10 771–10 780, 2018.

[12] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned image compression with discretized gaussian mixture likelihoods and attention modules," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7939–7948.

[13] T. Chen, H. Liu, Z. Ma, Q. Shen, X. Cao, and Y. Wang, "End-to-end learnt image compression via non-local attention optimization and improved context modeling," *IEEE Transactions on Image Processing*, vol. 30, pp. 3179–3191, 2021.

[14] J. Ascenso and E. Upenik, "White paper on jpeg ai scope and framework v1. 0," *ISO/IEC JTC 1/SC 29/WG1 N90049*, 2021.

[15] "Call for evidence for video coding for machines," *ISO/IEC JTC 1/SC 29/WG 2*, 2020.

[16] H. Choi and I. V. Bajić, "Scalable image coding for humans and machines," *IEEE Transactions on Image Processing*, vol. 31, pp. 2739–2754, 2022.

[17] J. Lainema, M. M. Hannuksela, V. K. M. Vadakital, and E. B. Aksu, "Hevc still image coding and high efficiency image file format," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 71–75.

[18] J.-R. Ohm and G. J. Sullivan, "Versatile video coding–towards the next generation of video compression," in *Picture Coding Symposium*, vol. 2018, 2018.

[19] S. Ma, X. Zhang, C. Jia, Z. Zhao, S. Wang, and S. Wanga, "Image and video compression with neural networks: A review," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[20] D. Mishra, S. K. Singh, and R. K. Singh, "Deep architectures for image compression: a critical review," *Signal Processing*, vol. 191, p. 108346, 2022.

[21] S.-P. Lu, R. Wang, T. Zhong, and P. 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. 10 816–10 825.

[22] J. Jing, X. Deng, M. Xu, J. Wang, and Z. Guan, "Hinet: Deep image hiding by invertible network," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 4733–4742.

[23] X. Luo, R. Zhan, H. Chang, F. Yang, and P. Milanfar, "Distortion agnostic deep watermarking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 548–13 557.

[24] X. Luo, Y. Li, H. Chang, C. Liu, P. Milanfar, and F. Yang, "Dvmark: A deep multiscale framework for video watermarking," *arXiv preprint arXiv:2104.12734*, 2021.

[25] E. Wengrowski and K. Dana, "Light field messaging with deep photographic steganography," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 1515–1524.

[26] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018.

[27] J. Bégaint, F. Racapé, S. Feltman, and A. Pushparaja, "Compressai: a pytorch library and evaluation platform for end-to-end compression research," *arXiv preprint arXiv:2011.03029*, 2020.

[28] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[29] R. C. Bose and D. K. Ray-Chaudhuri, "On a class of error correcting binary group codes," *Information and control*, vol. 3, no. 1, pp. 68–79, 1960.

[30] K. Choi, K. Tatwawadi, A. Grover, T. Weissman, and S. Ermon, "Neural joint source-channel coding," in *International Conference on Machine Learning*. PMLR, 2019, pp. 1182–1192.

[31] "Kodak photocd dataset," http://r0k.us/graphics/kodak/.

[32] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 7 2017.

[33] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3730–3738.

[34] B. Boehm, "Stegexpose-a tool for detecting lsb steganography," *arXiv preprint arXiv:1410.6656*, 2014.

[35] D. Lerch-Hostalot and D. Megías, "Unsupervised steganalysis based on artificial training sets," *Engineering Applications of Artificial Intelligence*, vol. 50, pp. 45–59, 2016.

[36] 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.