Deep-NC: A Secure Image Transmission Using Deep Learning and Network Coding

Quoc-Tuan Vien, Tuan T. Nguyen, Huan X. Nguyen

Abstract

Visual communications have played an important part in our daily life as a non-verbal way of conveying information using symbols, gestures and images. With the advances of technology, people can visually communicate with each other in a number of forms via digital communications. Recently Image Super-Resolution (ISR) with Deep Learning (DL) has been developed to reproduce the original image from its low-resolution version, which allows us to reduce the image size for saving transmission bandwidth. Although many benefits can be realised, the image transmission over wireless media experiences inevitable loss due to environment noise and inherent hardware issues. Moreover, data privacy is of vital importance, especially when the eavesdropper can easily overhear the communications over the air. To this end, this paper proposes a secure ISR protocol, namely Deep-NC, for the image communications based on the DL and Network Coding (NC). Specifically, two schemes, namely Per-Image Coding (PIC) and Per-Pixel Coding (PPC), are designed so as to protect the sharing of private image from the eavesdropper. Although the PPC scheme achieves a better performance than the PIC scheme for the entire image, it requires a higher computational complexity on every pixel of the image. In the proposed Deep-NC, the intended user can easily recover the original image achieving a much higher performance in terms of Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) than those at the eavesdropper. Simulation results show that an improvement of up to 32 dB in the PSNR can be obtained when the eavesdropper does not have any knowledge of the parameters and the reference image used in the mixing schemes. Furthermore, the original image can be downscaled to a much lower resolution for saving significantly the transmission bandwidth with negligible performance loss.

Q.-T. Vien and H. X. Nguyen are with the Faculty of Science and Technology, Middlesex University, United Kingdom. Email: {Q.Vien; H.Nguyen}@mdx.ac.uk.

T. T. Nguyen is with School of Computer & Mathematical Sciences, University of Greenwich, United Kingdom. Email: Tuan.Nguyen@greenwich.ac.uk.

Index Terms

Image communication; Deep learning; Super-resolution; Network coding

I. INTRODUCTION

Nowadays, personal digital images are widely exchanged over internet. In order to save the transmission bandwidth, an original high-resolution (HR) image can be converted into a low-resolution (LR) image prior to transmission, while image super resolution (ISR) can be employed at the receiver to recover the HR image from the LR image [1]. There is a need to protect these images due to privacy and essential information. Various methods have been developed to protect them such as stenography, encryption and watermarking. Stenography is a method to hide data inside an image, while in the encryption method, one image is converted into an encrypted image by using the secret key. On the other hand, watermarking techniques are to embed the signature into an image to visualise or hide the ownership of the image. In the following sections, we discuss several methods in this theme.

In general, the approaches of steganography and encryption are different, but they serve one purpose, protecting necessary data from irrelevant people. Encryption is more flexible and secure. Human, however, is always curious and tries to see what messages when they receive encrypted information. In such cases, steganography is a better option because it does not draw attention.

Network coding (NC) concept [2] has been well exploited in a vast number of research work from networking and communications perspective aiming to increase the system throughput by allowing the intermediate nodes to perform encoding the incoming data rather than operating simply as store-and-forward switches. Specifically, an algebraic approach of NC, namely random linear NC (RLNC), was developed in [3] where the intermediate nodes can perform random linear operations on the incoming data packets from different transmission source nodes. The sink nodes with a sufficient number of mixed data packets along with all RLNC coefficients can recover the data of all source nodes. The RLNC with its advantage in providing a dramatically enhanced throughput has been applied in a variety of network models, such as relaying [4], [5], multicasting [6], broadcasting [7] and multimedia streaming [8].

Considering data transfer over the wireless networks, several issues encounter during transmission such as data privacy and the amount of data transferred to reduce the time it takes to complete data transfer. Inspired by NC techniques, this work aims to develop a simple yet efficient

method to conceal and reduce the file size whilst transferring image data through network. The main contributions of the paper can be summarised as follows:

i) A secure ISR using both deep learning (DL) and NC, namely Deep-NC, is proposed for image communications between Alice and Bob. In the proposed scheme, an LR version of the original HR image is firstly generated at Alice using a simple bicubic interpolation. This downscaled image is then incorporated with the reference image by RLNC encoding prior to transmitting to Bob. Over noisy channel, the received image at Bob is denoised, followed by RLNC decoding using the shared reference image. Finally, ISR is applied to recover the original HR image. The novelty of the proposed scheme lies in the fact that it effectively utilises and links the strength of the NC and ISR into an unified framework to encode and decode the LR images at Alice and Bob, respectively.

ii) Two schemes, namely per-image coding (PIC) and per-pixel coding (PPC), are proposed. Specifically, the RLNC is applied in the PIC scheme for the entire image and the PPC scheme employs the RLNC on individual pixels of the image. Extensive experiments show that the PPC scheme, with the RLNC encoding and decoding at pixel level, achieves a better quality than the PIC scheme, while the PIC scheme requires a lower computational complexity with a lower bandwidth to process the whole image with only an RLNC coefficient.

iii) The performance of the proposed Deep-NC is evaluated taking into account two noise models including additive Gaussian noise and salt-and-pepper impulse noise as a typical non-Gaussian noise model. Simulation results show that Bob achieves a far better performance than Eve with the employment of Deep-NC, especially when Eve has no knowledge of the reference image shared between Alice and Bob. Additionally, both ISR and bicubic interpolation are shown to provide a better performance at Bob compared to Eve over both noise models. Furthermore, the proposed Deep-NC allows the image to be downscaled to a much lower resolution to save the transmission bandwidth, while still maintaining a significantly higher performance than Eve.

The rest of this paper is organised as follows: Section II firstly presents the background of DL and ISR. The related works of ISR techniques and secure image communications is then reviewed in Section III. Section IV describes the system model of a typical secure image communication over a noisy channel experiencing either additive Gaussian noise or salt-and-pepper noise. The

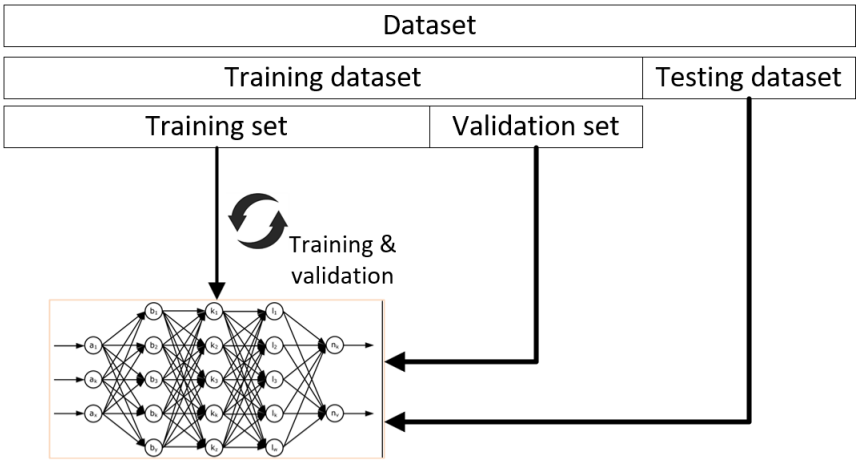

Fig. 1: Training and Testing stages in Deep learning.

proposed Deep-NC for secure ISR is presented in Section V, followed by the discussion of evaluation method in Section VI. Simulation results are presented in Section VII to validate the performance of the proposed Deep-NC. Section VIII finally concludes this paper summarising its main contributions with proposal of future works.

II. Background

A. Deep learning

The reasons for the success of DL can be attributed to two folds: first, the feature learning capacity of its hierarchical architecture allows for automatically extracting meaningful features from data [9], in which lower layers identify basic features and deeper layers synthesize higher-level features in terms of learned lower-level ones; second, the development of high-performance computers with Graphic Processing Units and Tensor Processing Units enable us to perform a large number of operations in reduced CPU time. Moreover, with the increasing digitisation transformation, the applicability of the DL is more pronounced as the bigger data becomes, the larger DL architecture is expected; and vice versa, to capture the underlying patterns better.

Figure. 1 illustrates the training and testing in DL which consist of the following steps:

i) Data collection: this step is to identify and collect relevant data such as images contain cats and images contain dogs. Each data is assigned by a specific label.

ii) Data pre-processing:

- Data collected might be noisy, missing data, extremely large values, unorganized text data. For example, images can contain other objects other than "cat". Cropping steps are needed to remove these artefacts.
- Data augmentation step might be needed to increase numbers of data. In term of images, it increases the number of images in the dataset by various transformations like rotation, flipping, colour variation, etc.
- The dataset is then splitted into the training and testing dataset.

iii) Building a DL architecture: creates a structure of the DL structure such as number of layers, how many nodes per layer.

iv) Training & Validating the model:

- The training dataset is further splitted into actual training dataset and validation dataset. The actual training dataset is fitted into the DL architecture to perform the training part.
- The validation set treated as unseen data is used to tune the hyper-parameters of the architecture. Hyper-parameters are specific values that are initialised before training the model.
- Features are automatically extracted from DL training. Whilst in traditional machine learning, features of data has to be specified manually for the algorithms.

v) Evaluating the model: the testing data is only used once a model is completely trained. The DL trained model extracted features from the testing data and then pass them through several layers to give the final prediction at the last layer.

B. Super Resolution for Image Processing

Image Super Resolution (ISR) is the process of recovering a high-resolution (HR) image from a given low-resolution (LR) image. The LR image is an image which has a reduced dimension or contains noise or blurring regions. The relationship between an HR and an LR image can be represented by a function $HR = F(LR, \gamma)$ whereas F is the function used to invert the LR image back the HR image and γ is the parameter for the inversion process.

If we know function F, the HR image can be easily reconstructed from the LR image. In reality, this is, however, not the case. Research has proven that DL is one of effective techniques for ISR tasks.

The core problem of ISR is to create a mapping between the LR and HR images. In order to do so, multiple DL based architectures have been developed. These architectures can be grouped into four main architecture types as shown in Fig. 2 [1] including:

i) Pre-upsampling super resolution: As visualised in Fig. 2(a), the LR image is firstly upsampled by traditional upsampling methods, e.g. bicubic [10] and bilinear interpolation [11]. Thus, a Super-Resolution Convolutional Neural Network (SRCNN) was proposed in [12] to learn an end-to-end mapping from interpolated LR images to HR images. This type of method can produce blurring regions due to the predefined upsampling.

ii) Post-upsampling: This approach performs upsampling at the last learnable layer as shown in Fig. 2(b). With this strategy, the feature extraction is performed in the lower dimensional spaces. Hence, it reduces the complexity.

iii) Progressive Upsampling: The above two techniques are using one upsampling convolution which makes difficulty in learning for larger scales. To address this, a progressive upsampling architecture adopted from Laplacian pyramid SR framework, namely LapSRN, was introduced in [13]. As shown in Fig. 2(c), images are upsampled to higher resolution at each stage. Models for these frameworks are in general complex due to multiple stages.

iv) Iterative Up and Down Sampling: In order to capture coherence between LR and HR image pair, the back-projection [14] was exploited and utilised in ISR to iteratively refine intensity of the HR image as shown in Fig. 2(d). Several methods, e.g. Deep Back-Projection Networks (DBPN) [15] and Super-Resolution FeedBack Network (SRFBN) [16], were introduced. These models can provide high quality reconstruction results but designing criteria for back-projection still needs further exploration.

III. Related Works

A. Super Resolution

Kim et al. [17] proposed a very deep convolutional network, namely Very Deep Super Resolution (VDSR), for generating an HR image given its LR variants which achieve high benchmark scores with relatively high speed against other top-performing methods. Fig. 3 shows an overview of DL architecture, in which each convolutional layer contains 64 filters and the rectified linear units (ReLu) are utilised as an activate function.

By leveraging the residual learning strategy and batch normalisation on top of a very deep Convolutional Neural Network (CNN) architecture the method achieves competitive results in

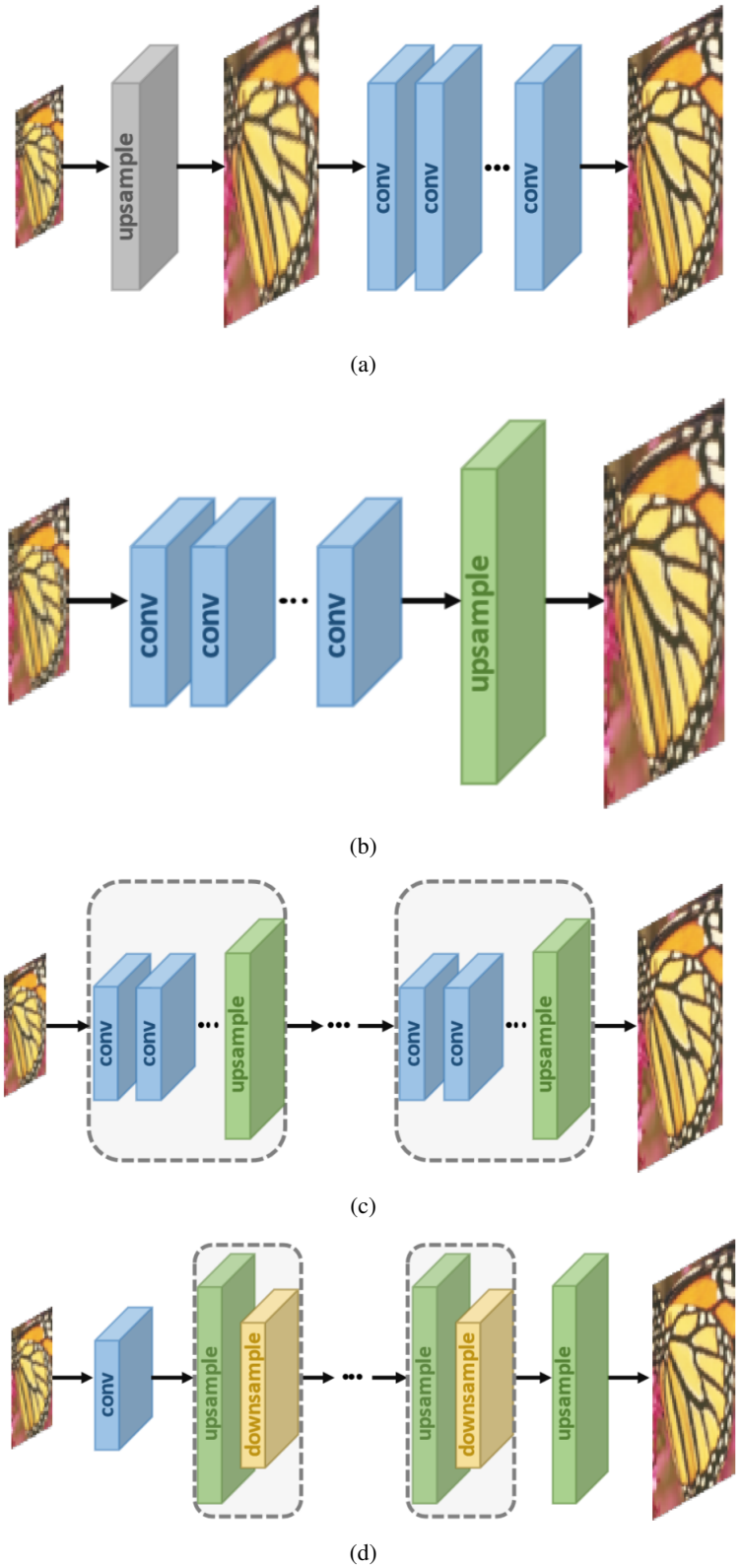

Fig. 2: Super resolution architectures [1]: (a) Pre-upsampling; (b) Post-upsampling; (c) Progressive Upsampling; and (d) Iterative Up and Down Sampling

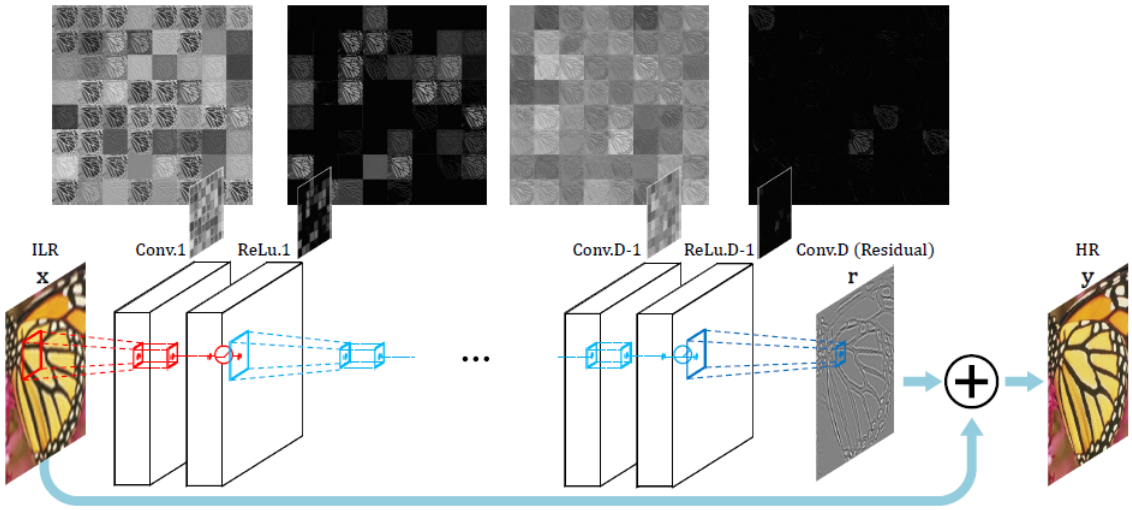

Fig. 3: VDSR framework [17]

denoising with appealing run time. That is why since its introduction, it has been adopted in various fields, such as seismic [18], machinery [19], as well as network security [20].

B. Encryption

Guerrini et al. [21] investigated a secure image hash-based geometric transformation procedure for transmitting an image with low information knowable to ensure security purposes. In addition, the method showed to be robust to some degrees of transformations such as rotation, shifting, and scaling. Peng et al. [22] utilised the optimised semi tensor product CS algorithm coupling with the digital watermarking algorithm for ensuring security and effectiveness of Telemedicine Image transmission. The experimental results can achieve reconstructed imaged with high quality up to around 33 dB PSNR; moreover, good visual images still obtained with the presence of different noises including Gaussian white noise, Salt and Pepper noise, and Poisson noise. Xie et al. [23] explored an $n \times n$ secret image sharing scheme, which ensures the security during transmission based on the principle that the host images can be retrieved perfectly if and only if n shares work together. In 2017, Zhang et al. [18] proposed a denoising approach based on CNNs, a.k.a DnCNNs, effectively coping with the presence of unknown level Gaussian noises in concerned images.

C. Stenography

Stenography can be applied into different types of file formats, such as text, image, video, and audio. Images are most suitable for stenography as they consist of large amount of redundant bits. A stego image is obtained when a message is embedded into a cover image. The stego image is sent through a public channel. Audio and video files consist of similar techniques as for image files. In this section, several methods for image stenography will be reviewed.

A popular and easy technique is the Least Significant Bit (LSB) used in [24], [25]. It uses the last bit of each pixel to store the secret message's data bit. This method is simple yet less robust. The hidden data can be lost easily when the image is manipulated. In other hand, Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT), used in [26] and [27] respectively, transform an image from the spatial domain to the frequency domain. They, then, separate the image into low, middle and high frequency. The high frequency is normally removed through compression and the low frequency contains the most important visual parts of the image. The hidden message is embedded by modifying the middle frequency so that the change of image will not be noticed. The problem with this technique is that the hidden message must have smaller size than the cover image, which means that hiding an image into another image is not easy to be achieved.

Inspired by these methods, our work combines them into a unified framework to secure images transmitted in the network. It is discussed in the next section.

IV. System Model

Figure 4 illustrates a typical secure image communication model where Alice ($\mathcal{A}$) wants to send a private image to Bob ($\mathcal{B}$) in the existence of Eve ($\mathcal{E}$) who is trying to eavesdrop the image.

Let I_A denote the original image of size $M \times N$ that Alice wants to send. Considering typical colour images with three channels, i.e. red, green and blue, I_A can be defined as an $M \times N \times 3$ array, i.e.

$$I_A : f(x_A, y_A) \rightarrow \mathbb{R}^3, \tag{1}$$

where $f(x_A, y_A) \in \mathbb{R}$ is the intensity of the image pixel at point (x_A, y_A).

Over the transmission media, the noise is inevitable, which causes image degradation. Generally, Gaussian noise model has been regarded as the best fit in representing the additive noise in the undesired signal at the receiver in most of communication systems due to its simplicity

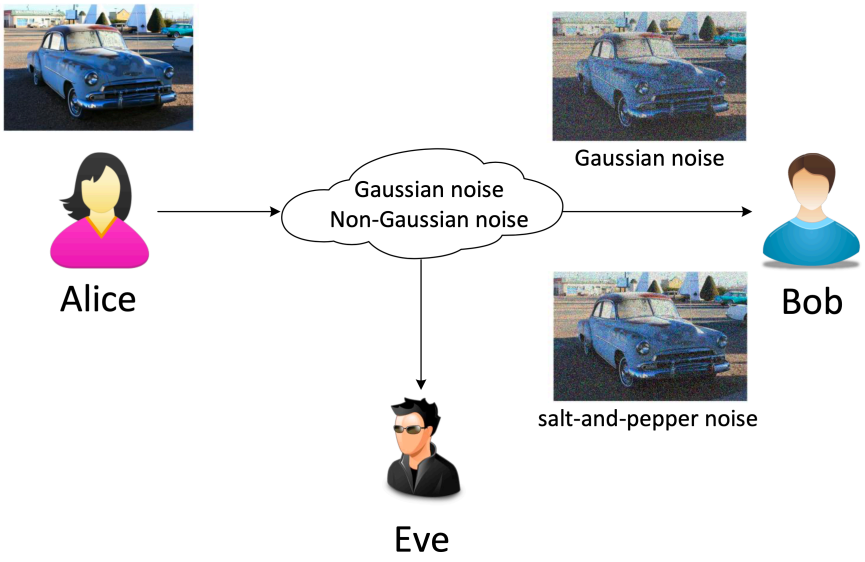

Fig. 4: System model of secure image transmission.

with dominant central limit theorem. However, there are some other non-Gaussian noise models, such as impulse noise and Cauchy noise, which are used to model image degradation caused by malfunction of the image detector or other hardware issues. Specifically, in this paper, two types of noise model are considered, including Gaussian noise and impulse noise in the form of salt-and-pepper noise.

A. Additive Gaussian Noise

Considering additive noise model, the image received at node $\mathcal{X}$, $\mathcal{X} \in \{\mathcal{B}, \mathcal{E}\}$, is given by

$$I_X^{(G)} = I_A + N_X^{(G)}, \tag{2}$$

where $N_X^{(G)}$, $X \in \{B, E\}$, is additive white Gaussian noise (AWGN) at $\mathcal{X}$ having mean of μ_X and variance of σ_X^2.

In order to denoise an image with AWGN, deep neural network can be employed using a pretrained DnCNN network [18].

B. Salt-and-Pepper Noise

The salt-and-pepper noise is represented by noise density which is defined as the probability of having noise in the received image. Let d, $0 \leq d \leq 1$, and $I_X^{(SP)}$ denote the salt-and-pepper noise density and the received image at node $\mathcal{X}$, $\mathcal{X} \in \{\mathcal{B}, \mathcal{E}\}$, respectively. The probability of

receiving the original image at node $\mathcal{X}$ is thus $\Pr[I_X^{(SP)} = I_A] = 1 - d$. In other words, $I_X^{(SP)}$ can be written by

$$I_X^{(SP)} = \begin{cases} N_X^{(SP)} & \text{with probability } d, \\ I_A & \text{with probability } (1-d), \end{cases} \tag{3}$$

where $N_X^{(SP)}$ is the salt-and-pepper noise at node $\mathcal{X}$.

The salt-and-pepper noise can be effectively treated by applying median filter which is widely used as a non-linear approach to remove impulse noise in the image [28].

V. Proposed Secure Image Super-Resolution with Deep Network Coding

In the proposed secure ISR, the original HR image at Alice is firstly downscaled to an LR image. Then, RLNC is employed to encode this downscaled image prior to transmitting to Bob over the noisy channel which can be overheard by Eve.

At Bob, the received image is firstly denoised, and then it is decoded with RLNC decoding. The original image can be recovered using ISR. A similar image processing is employed at Eve, apart from the RLNC decoding which needs to be estimated since Eve does not have any knowledge of the RLNC encoding scheme at Alice.

The flowchart of the proposed secure ISR is shown in Fig. 5, which consists of the following main steps:[1]

A. Downscaling at Alice

In order to save the transmission bandwidth, an LR version of the original image is firstly generated using bicubic interpolation. The bicubic filter is employed for downscaling due to its low computational complexity, while preserving image details with smooth interpolated surface.[2]

After downscaling the original image, i.e. I_A, from Alice, we have an LR image I'_A having size $\lceil M/\delta \rceil \times \lceil N/\delta \rceil \times 3$ where δ denotes the scaling factor and $\lceil x \rceil$ denotes the ceiling function of x.

[1]Note that the common blocks at Bob and Eve to represent the same steps are combined in the following discussion.

[2]There exist different downscaling methods, e.g. Box sampling, Nearest-neighbour interpolation, Lanczos Filtering [29], or CNN based downscaling like [30]; however, we do not need a sophisticated downscaling method in this work.

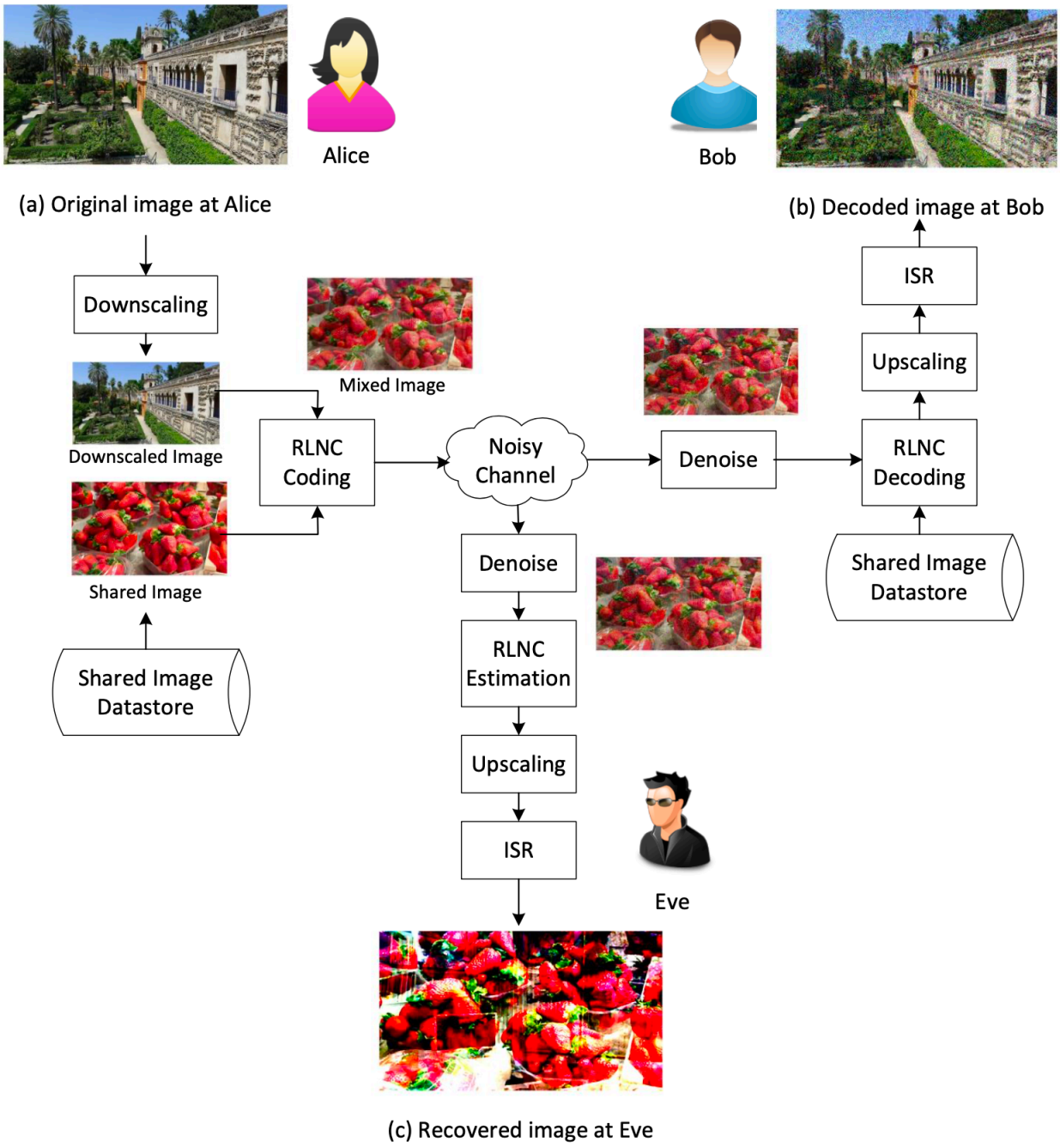

Fig. 5: Proposed Deep-NC for secure image super-resolution

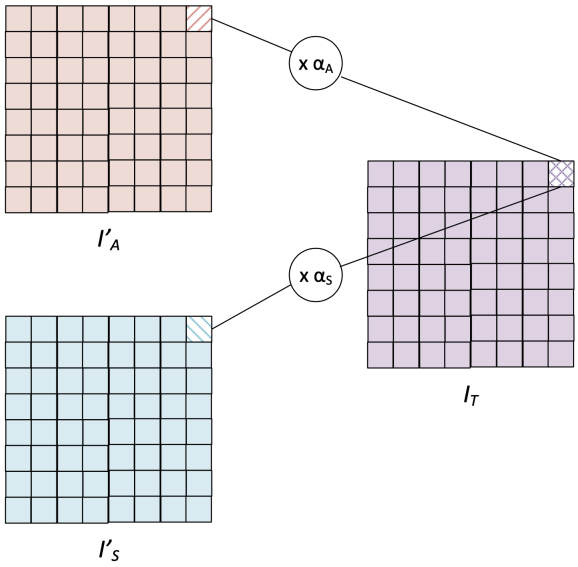

Fig. 6: RLNC encoding at Alice

B. RLNC Encoding at Alice

Following the concept of RLNC [3], in the proposed scheme, the downscaled image at Alice is linearly mixed with a reference image using random scalar coefficients, a.k.a. RLNC coefficients. As shown in Fig. 5, the reference image is acquired from an image datastore which is assumed to be shared between users as a common image dataset.[3]

In order to mix two images, the shared image is first downscaled to the same size of the original image. Let I_S and I'_S denote the HR and LR versions of the shared image, respectively. I'_S should have the same size of I'_A, i.e. an $\lceil M/\delta \rceil \times \lceil N/\delta \rceil \times 3$ array.

The mixing of the original image and the shared image can be realised via either per-image coding (PIC) or per-pixel coding (PPC) scheme as follows:

1) Proposed Per-Image Coding (PIC): In the PIC scheme, the RLNC is applied for the entire images with the same RLNC coefficients for all pixels of the original and shared images. As illustrated in Fig. 6, the transmitted image at Alice, denoted by $I_T^{(PIC)}$, can be written by

$$I_T^{(PIC)} = \alpha_A I'_A + \alpha_S I'_S, \tag{4}$$

[3]Notice that only the RLNC coeffients and the index of the reference image are shared between Alice and Bob prior to transmission as private keys in key-agreement protocols.

where α_A and α_S are RLNC coefficients having $\alpha_A > 0$, $\alpha_S \geq 0$ and $\alpha_A + \alpha_S = 1$. Here, α_A and α_S represent the fractions of original image and shared image, respectively, in the mixed image.

2) Proposed Per-Pixel Coding (PPC): Instead of considering the whole image, the mixed image at Alice in the PPC scheme is generated by treating the pixels in both original and shared images as

$$I_T^{(PPC)} = \mathbf{M}_A \circ I'_A + \mathbf{M}_S \circ I'_S, \tag{5}$$

where $\mathbf{M}_A$ and $\mathbf{M}_S$ are matrices of RLNC coefficients having size $\lceil M/\delta \rceil \times \lceil N/\delta \rceil \times 3$, and $\circ$ denotes the Hadamard product of two matrices. Similar to the condition of the RLNC coefficients in the PIC scheme, the random matrices $\mathbf{M}_A$ and $\mathbf{M}_S$ are also required to be $\mathbf{M}_A > 0$, $\mathbf{M}_S \geq 0$ and $\mathbf{M}_A + \mathbf{M}_S = 1$.

Remark 1 (Image Decodability-Security Tradeoff)**.** *In the PIC scheme, a higher α_A results in a better performance at Bob with an enhanced image decodability, while a higher α_S helps secure the original image from Eve. In fact, a higher α_A means more information of the original image in the mixed image, and thus Bob can recover the desired image with higher decodability. However, Eve also overhears the same amount of information to be able to extract the original image, which means more information at the same time is leaked to Eve. On the other hand, a higher α_S, i.e. a lower α_A, causes degradation of the image decodability at Bob, but it helps improve the secrecy of the image communications due to less information of the original image in the image mixture. Therefore, it is crucial to find α_A (or α_S) to balance the tradeoff between the image decodability and security. The impacts of RLNC coefficients on the performance of the proposed scheme are discussed in Subsection VI-B.*

Remark 2 (Comparison of PIC and PPC Schemes)**.** *It can be noticed that the PIC is a special case of the PPC scheme when all RLNC coefficients in matrices $\mathbf{M}_A$ and $\mathbf{M}_S$ are identical. Although the PIC is shown to be much simpler than the PPC scheme, they both have their own merits and demerits as follows:*

- *PIC scheme: requires low computational complexity with low bandwidth to share only an RLNC coefficient, i.e. α_A or α_S. However, Eve can easily estimate this RLNC coefficient in the range between 0 and 1.*
- *PPC scheme: involves matrix computation at both transceivers, and thus it requires a high computational complexity. Also, in the PPC scheme, a high bandwidth is required to share the matrix of RLNC coefficients, i.e. $\mathbf{M}_A$ (or $\mathbf{M}_S$), especially when the size of the original image is large. However, in return for this, an enhanced performance is obtained due to the fact that all pixels of the original image are encrypted with different RLNC coefficients.*

C. Denoise at Bob and Eve

Over the noisy channel, the image transmitted from Alice is deteriorated, which is caused by either Gaussian or non-Gaussian noises. In this subsection, Gaussian noise and salt-and-pepper

noise models are sequentially presented, followed by the introduction of denoising techniques employed at Bob and Eve with the assumption that Eve experiences the same noisy environment as Bob.

1) Gaussian Noise: The images received at $\mathcal{B}$ and $\mathcal{E}$ over Gaussian noise model can be obtained as in (2), i.e.

$$I_X^{(G)} = I_T + N_X^{(G)}, \tag{6}$$

where $X \in \{B, E\}$ and I_T is the transmitted image at Alice given by (2). Both $\mathcal{B}$ and $\mathcal{E}$ then remove the Gaussian noise using the same pretrained DnCNN network.

2) Salt-and-Pepper Noise: As shown in (3), the images received at $\mathcal{B}$ and $\mathcal{E}$ caused by salt-and-pepper noise can be given by

$$I_X^{(SP)} = \begin{cases} N_X^{(SP)} & \text{with probability } d, \\ I_T & \text{with probability } (1-d), \end{cases} \tag{7}$$

where $X \in \{B, E\}$ and d is salt-and-pepper noise density. The salt-and-pepper noise can be filtered out by employing median filter.

For brevity, let us denote the denoised images at $\mathcal{B}$ and $\mathcal{E}$ after filtering by $\tilde{I}_B$ and $\tilde{I}_E$, respectively.

D. RLNC Decoding at Bob

The RLNC decoding at Bob can be carried out with respect to two RLNC encoding schemes (see Subsection V-B) as follows:

1) PIC Decoding: Given the index of the reference image in the shared image datastore and RLNC coefficients, i.e. α_A or α_S, Bob can recover the image transmitted from Alice as

$$\hat{I}_B^{\prime(PIC)} = \frac{1}{\alpha_A}\left(\tilde{I}_B - \alpha_S I_S'\right), \tag{8}$$

where I_S' is the LR version of the shared image and $\hat{I}_B^{\prime(PIC)}$ is the decoded image at $\mathcal{B}$ in the PIC scheme.

2) PPC Decoding: For PPC decoding, a matrix of RLNC coefficients, i.e. $\mathbf{M}_A$ or $\mathbf{M}_S$, is required at Bob. The original image transmitted from Alice can be decoded as

$$\hat{I}_B^{\prime(PPC)} = \left(\tilde{I}_B - \mathbf{M}_S \circ I_S'\right) \oslash \mathbf{M}_A, \tag{9}$$

where $\oslash$ denotes the Hadamard division of two matrices and $\hat{I}_B^{\prime(PPC)}$ denotes the decoded image at $\mathcal{B}$ in the PPC scheme.

For clarity, the RLNC encoding at Alice and decoding at Bob are summarised in Algorithms 1 and 2, respectively.

Algorithm 1 RLNC encoding at Alice

Input: A reference image I_S, downscaled original image I'_A, and scaling factor δ

1: Downscale $I_S \rightarrow I'_S$ having the same size of I'_A, i.e. $\lceil M/\delta \rceil \times \lceil N/\delta \rceil \times 3$
2: Generate a random coefficient α_A (PIC scheme) or a random matrix $\mathbf{M}_A$ (PPC scheme)
3: Mix the downscaled original image I'_A and I'_S using RLNC encoding:
PIC scheme: $I_T^{(PIC)} = \alpha_A I'_A + \alpha_S I'_S$
PPC scheme: $I_T^{(PPC)} = \mathbf{M}_A \circ I'_A + \mathbf{M}_S \circ I'_S$

Output: Transmitted image: $I_T^{(PIC)}$ or $I_T^{(PPC)}$

Algorithm 2 RLNC decoding at Bob

Input: A reference image I_S, denoised image $\tilde{I}_B$, and RLNC coefficients α_A (PIC scheme) or RLNC coefficient matrix $\mathbf{M}_A$ (PPC scheme)

1: Downscale $I_S \rightarrow I'_S$ having the same size of $\tilde{I}_B$
2: Separate the received image $\tilde{I}_B$ using RLNC decoding:
PIC scheme: $\hat{I}_B^{\prime(PIC)} = \frac{1}{\alpha_A}\left(\tilde{I}_B - \alpha_S I'_S\right)$
PPC scheme: $\hat{I}_B^{\prime(PPC)} = \left(\tilde{I}_B - \mathbf{M}_S \circ I'_S\right) \oslash \mathbf{M}_A$

Output: Decoded image: $\hat{I}_B^{\prime(PIC)}$ or $\hat{I}_B^{\prime(PPC)}$

E. RLNC Estimation at Eve

Trying to decode the image shared from Alice, Eve however does not know the RLNC coefficients, i.e. α_A and α_S, and the index of the image in the shared datastore, i.e. I_S, which was used at Alice for mixing it with the original image. Therefore, Eve has to estimate the RLNC parameters and select the right image in the datastore to recover the image.

The RLNC coefficients estimated at Eve can be written by

$$\hat{\alpha}_A = \alpha_A \pm \epsilon_A, \tag{10}$$

$$\hat{\alpha}_S = 1 - \hat{\alpha}_A, \tag{11}$$

where $\hat{\alpha}_A$ and $\hat{\alpha}_S$ are estimated RLNC coefficients of α_A and α_S, respectively, and ϵ_A denotes the estimation error of α_A at Eve knowing the fact that $\hat{\alpha}_A + \hat{\alpha}_S = 1$.

In addition to the RLNC coefficients, Eve needs to predict the image in the shared datastore that was used for encoding at Alice. Letting $\hat{I}'_S$ denote the predicted image, Eve can decode the image following the same method at Bob in Subsection V-D for either PIC or PPC decoding.

F. ISR at Bob and Eve

To be able to recover original images from decoded images by RLNC, the ISR framework, mentioned in Subsection III-A, was implemented. The ISR was trained with a public available IAPR TC-12 Benchmark dataset [31]. The training used batches of size 64 with 100 epochs. The learning rate is initially set as 0.1 and decreased by a factor of 10 in every 10 epochs. The ISR was trained with a scaling factor $\delta = 4$. These hyper-parameters used are similar to the optimal ones reported by [17].

Then, the decoded images by RLNC decoding at Bob, $\hat{I}_B^{\prime(PIC)}$ or $\hat{I}_B^{\prime(PPC)}$, and at Eve, $\hat{I}_E^{\prime(PIC)}$ or $\hat{I}_E^{\prime(PPC)}$ are fed into ISR model to obtain the upscaled images at Bob and Eve, denoted by $\hat{I}_B$ and $\hat{I}_E$. respectively. The upscaled images can be written by

$$\hat{I}_B = ISR(\hat{I}_B^{\prime(PIC)}), \tag{12}$$

$$\hat{I}_E = ISR(\hat{I}_E^{\prime(PIC)}), \tag{13}$$

where $ISR(\cdot)$ can be thought as a function to reconstruct the images using ISR with the scaling factor δ.

VI. Evaluation Method of the Proposed Deep NC

In order to evaluate the effectiveness of the proposed Deep-NC for a secure ISR, Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) are sequentially presented in this section as two performance metrics to compare the quality of the recovered image at Bob and Eve, i.e. $\hat{I}_B$ and $\hat{I}_E$, with the original HR image transmitted from Alice, i.e. I_A.

A. Peak Signal-to-Noise Ratio

The peak signal-to-noise ratio (PSNR) is the ratio between a signal's maximum power and the power of the signal's noise. Engineers commonly use the PSNR to measure the quality of reconstructed images that have been compressed. Each picture element (pixel) has a color value that can change when an image is compressed and then uncompressed. Signals can have a wide dynamic range, so PSNR is usually expressed in decibels, which is a logarithmic scale.

As a well-known image comparison metric, the PSNR is considered to evaluate the loss of the image quality. In the proposed secure ISR, the loss is due to not only the noise at Bob, but also the lack of details in the LR image and the training loss in the ISR network.

The PSNR, in dB, of the recovered image $\hat{I}_B$ at Bob with respect to the original image I_A is defined as[4]

$$\text{PSNR} \triangleq 10 \log_{10} \frac{1}{\text{MSE}}, \tag{14}$$

where MSE is the mean square error between $\hat{I}_B$ and I_A given by

$$\text{MSE} \triangleq E\left[\left(I_A - \hat{I}_B\right)^2\right]. \tag{15}$$

Here, $E[\cdot]$ denotes the expectation operator.

Considering RGB colour images having size $M \times N$ with three RGB values per pixel, the MSE can be calculated by

$$\text{MSE} = \frac{1}{3MN} \sum_{x=1}^{M} \sum_{y=1}^{N} \sum_{z=1}^{3} \left(I_A(x,y,z) - \hat{I}_B(x,y,z)\right)^2. \tag{16}$$

Note that the PSNR at Eve is also evaluated using (14). However, the quality of the recovered image is further degraded due to the unknown reference image and the lack of the information of RLNC coefficients which are only shared between Alice and Bob.

B. Structural Similarity Index Measure

The SSIM is an image metric which is used to measure the similarity between two images based on their visible structure [32]. The SSIM, unlike the PSNR, evaluates the image distortion through the perceived quality difference between the two images, which is represented by a multiplicative combination of three factors of the images including luminance distortion, contrast distortion and structural correlation loss.

The per-pixel SSIM of the recovered image $\hat{I}_B$ at Bob with respect to the original image I_A at a pixel (x, y), $1 \leq x \leq M$, $1 \leq y \leq N$, is defined as

$$\text{SSIM}(x,y) \triangleq [l_{A,B}(x,y)]^{\alpha} [c_{A,B}(x,y)]^{\beta} [s_{A,B}(x,y)]^{\gamma}, \tag{17}$$

where $l_{A,B}(x,y)$, $c_{A,B}(x,y)$ and $s_{A,B}(x,y)$ denote the luminance, contrast and structure components, respectively, at the pixel (x,y), and α, β and γ denote the weighted combination of these three components correspondingly. For simplicity, it is generally assumed the luminance,

[4]In this work, the image is in double-precision floating-point data type having maximum possible pixel value of 1.

contrast and structure components have equal influence on the image, i.e. $\alpha = \beta = \gamma = 1$. Here, $l_{A,B}(x,y)$, $c_{A,B}(x,y)$ and $s_{A,B}(x,y)$ are computed by [32]

$$l_{A,B}(x,y) = \frac{2\mu_A\mu_B + C_1}{\mu_A^2 + \mu_B^2 + C_1}, \tag{18}$$

$$c_{A,B}(x,y) = \frac{2\sigma_A\sigma_B + C_2}{\sigma_A^2 + \sigma_B^2 + C_2}, \tag{19}$$

$$s_{A,B}(x,y) = \frac{\sigma_{AB} + C_3}{\sigma_A\sigma_B + C_3}, \tag{20}$$

respectively, where μ_A and μ_B are local mean; σ_A and σ_B are standard deviations; σ_{AB} is covariance of images I_A and $\hat{I}_B$; and C_1, C_2 and C_3 are regularization constants for the luminance, contrast, and structural components, respectively. Considering images of double-precision floating-point data type with pixel intensity in $[0,1]$, the constants C_1, C_2 and C_3 are set as $C_1 = 0.01^2$, $C_2 = 0.03^2$ and $C_3 = C_2/2$.

The mean SSIM (MSSIM) of the entire image $\hat{I}_B$ is therefore computed by

$$\text{MSSIM} = E\left[\text{SSIM}\right] = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}\text{SSIM}(x,y). \tag{21}$$

VII. Simulation Results

In this section, we present the simulation results of the proposed Deep-NC for secure ISR. Two performance metrics, i.e. PSNR and SSIM, are considered to validate the effectiveness of the two proposed PIC and PPC schemes. We first compare the performance at Bob with that at Eve when employing either bicubic interpolation or VDSR for converting the received LR images to the original HR images. The impacts of RLNC coefficients, shared image dataset, Gaussian noise, salt-and-pepper noise and scaling factor on the performance are then sequentially evaluated to show the enhancement of the proposed scheme in securing the private image from the Eve. The results are obtained by simulation in MATLAB. The training is performed on an image dataset of the IAPR TC-12 benchmark [31][5]. For validation of the proposed scheme, 20 undistorted images of the Image Processing Toolbox in MATLAB are used as shown in Fig. 7 in which the last image is selected as a reference image[6].

[5]The image dataset of the IAPR TC-12 benchmark consists of 20,000 still natural images which are available free of charge and without copyright restrictions.

[6]During our experiments, we notice that if Alice uses a reference image which contains a plain background and less clutter like the 2nd image from the datastore (see Fig. 7), the mixed image can show small amount of the original image. Those artefacts are not clear enough to be spotted without a proper glimpse.

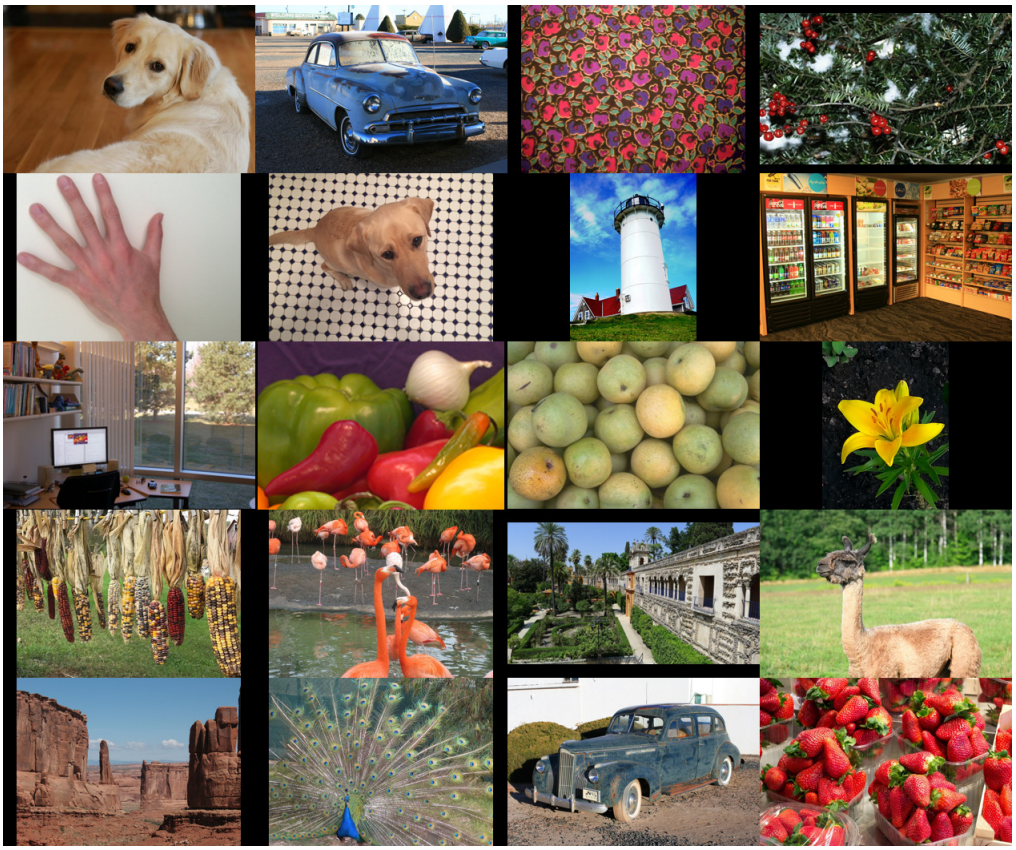

Fig. 7: Images for evaluation of the proposed Deep-NC.

It is worth to reinforce that below experiments to demonstrate effectiveness of our proposed methods. To the best of our knowledge, our work is the fist approach which combining NC and ISR to enhance data privacy and reduce the data transmission via network. Each component of our framework, mentioned in Fig. 5, can be replaced by sophisticated methods to increase the quality. The ISR component, for instance, can be replaced by RDN [33].

A. PIC versus PPC Schemes

Evaluating the performance of the proposed PIC and PPC schemes, Figs. 8 and 9 sequentially plot the PSNR and SSIM against the noise variance in Gaussian noise model. Two upscaling schemes, i.e. bicubic interpolation and VDSR, are considered at Bob and Eve. For a fair comparison, it is assumed that the AWGN at Bob and Eve has the same noise variance, i.e. $\sigma_B^2 = \sigma_E^2 = \sigma^2$. For RLNC encoding, the coefficients of the original image, i.e. elements in the RLNC coefficient matrix $\mathbf{M}_A$, are randomly selected in the range of $[0.3, 0.5]$. It is assumed that

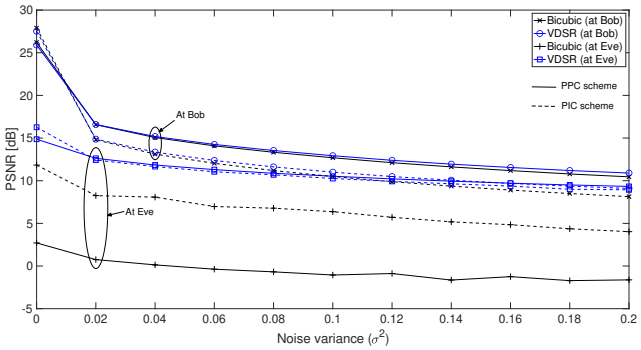

Fig. 8: PSNR versus noise variance with PIC and PPC coding schemes.

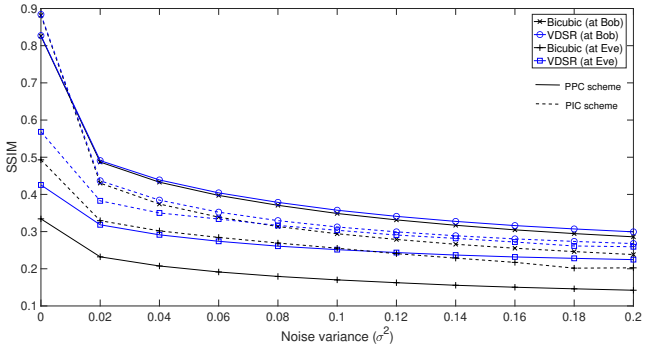

Fig. 9: SSIM versus noise variance with PIC and PPC coding schemes.

Eve is also aware of the reference image used at Alice. As shown in Fig. 8, Bob achieves a higher PSNR than Eve of up to 11 dB in both PIC and PPC schemes with VDSR upscaling in case of no noise. This is attained as a result of the secure RLNC encoding at Alice. It can be further observed that the PPC scheme helps not only increase the PSNR and SSIM at Bob, but also degrade the performance at Eve compared to the PIC scheme. This accordingly verifies the observations in Remark 2 regarding the outperformance of the PPC over the PIC scheme in exchange for a higher computational complexity. Moreover, an enhanced performance is achieved with the VDSR compared to the bicubic interpolation. Such enhancement can be clearly noticed at Eve when the VDSR is employed, though its performance is still lower than Bob. For instance, 12 dB in PSNR can be enhanced at Eve with the VDSR upscaling.

B. Impacts of RLNC coefficients

Investigating the impacts of RLNC coefficients on the performance of the proposed scheme, Fig. 10 plots the PSNR of the PIC scheme versus the RLNC coefficient of the original image, i.e.

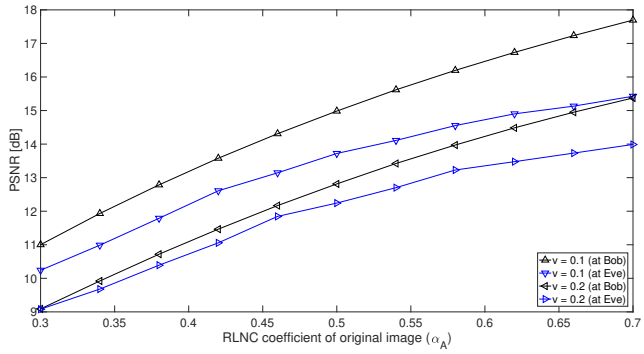

Fig. 10: PSNR versus RLNC coefficient of original image.

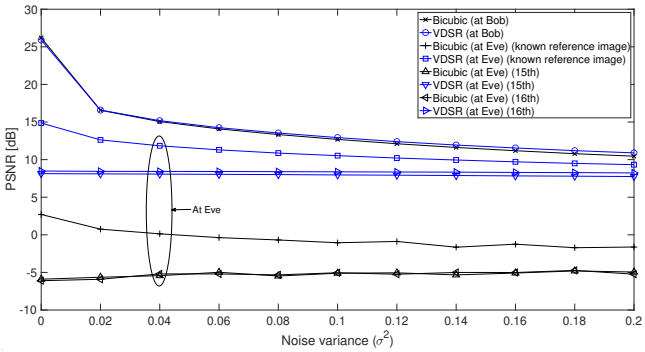

Fig. 11: PSNR versus noise variance with respect to different shared images at Eve.

α_A, with respect to different noise variances at Bob and Eve. Specifically, two noise variances, i.e. $\sigma_B^2 = \sigma_E^2 = \sigma^2 = \{0.1, 0.2\}$, are considered. It can be seen that Bob achieves a better performance than Eve for all RLNC coefficients. Also, the PSNR at both Bob and Eve increases as α_A increases. This is due to the fact that there is more information of the original image in the mixed image. As noted in Remark 1, the RLNC coefficients should be selected so as to restrict Eve from recovering the original image, while maintaining the higher image decodability at Bob. For instance, α_A should be less than 0.5 to limit the PSNR at Eve by 14 dB when $\sigma^2 = 0.1$. The PSNR coefficients should be thus selected depending also on the noise variance. As shown in Fig. 10, with the same requirement of the maximal PSNR at Eve by 14 dB, α_A should not be greater than 0.5 and 0.7 when $\sigma^2 = 0.1$ and $\sigma^2 = 0.2$, respectively.

C. Impacts of Shared Image Dataset

In order to decode the original image, apart from the RLNC coefficients, the reference image in the datastore is required to be known at the receiver. Considering the scenario when Eve may

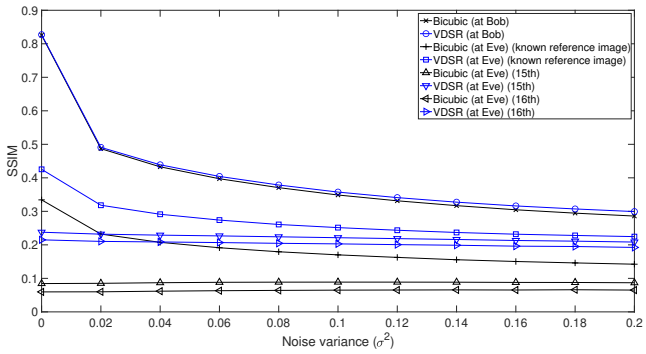

Fig. 12: SSIM versus noise variance with respect to different shared images at Eve.

not know the reference image used at Alice or may use the incorrect image, Figs. 11 and 12 sequentially plot the PSNR and SSIM of the proposed scheme as functions of the noise variance with respect to different shared images at Eve including the 15th, 16th and 20th reference images (see Fig. 7). Here, the 20th image is used for the RLNC encoding at Alice with PPC scheme. The other parameters are similarly set as in Figs. 8 and 9 with bicubic interpolation and VDSR for upscaling. It can be observed in both Figs. 11 and 12 that the PSNR and SSIM at Eve are considerably degraded when the wrong reference image is selected for decoding. For instance, in the noise-free environment with VDSR upscaling, the PSNR at Eve decreases by 7 dB when using the 15th or 16th image instead of the 20th image. A further notice is the fact that, without knowledge of the RLNC coefficient matrix, Eve can only achieve a close performance to Bob even with the right reference image. This accordingly reflects the effectiveness of the proposed scheme in securing the original image. Furthermore, as shown in Fig. 11, Bob achieves a much better performance than Eve of up to 32 dB when Eve implements only bicubic interpolation. This is due to the fact that no ISR is involved in the image processing at Eve in this case along with estimation error of RLNC coefficient matrix and the unknown reference image.

D. Impacts of Gaussian Noise

Considering AWGN model in image communication over wireless medium, Figs. 13 and 14 plot the PSNR and SSIM of the proposed PPC scheme versus noise variance, i.e. σ^2 with the assumption that Bob and Eve experience the same noise model. Two upscaling schemes, i.e. bicubic interpolation and VDSR, are employed at Bob and Eve. The reference image shared to Bob is also leaked to Eve. Intuitively, it can be seen that both the PSNR and SSIM decrease as the noise variance increases. Over the wireless medium, the VDSR is also shown to be beneficial

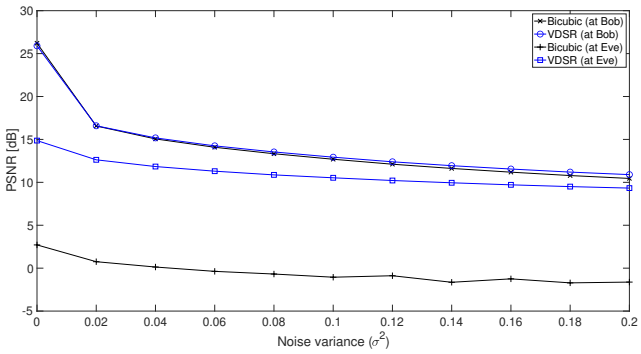

Fig. 13: PSNR versus noise variance in Gaussian white noise model.

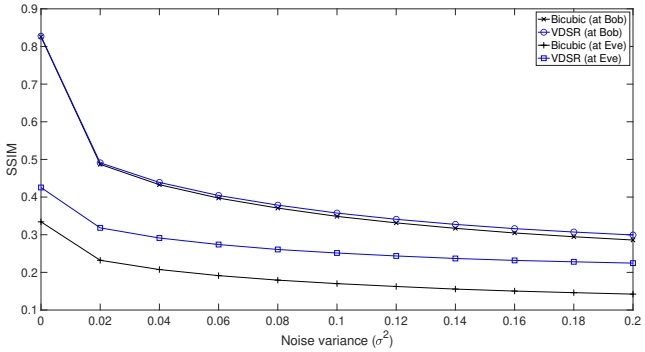

Fig. 14: SSIM versus noise variance in Gaussian white noise model.

providing higher PSNR and SSIM than the bicubic interpolation. Similarly, Bob is shown to achieve a better performance than Eve over the whole range of noise variance with both bicubic interpolation and VDSR, though the performance gap is smaller in the more lossy environment. For instance, with the VDSR upscaling, the PSNR at Bob is 3 dB and 2 dB higher than that at Eve, while a considerable enhancement of 14 dB and 13 dB can be achieved with the bicubic interpolation when $\sigma^2 = 0.1$ and $\sigma^2 = 0.2$, respectively.

E. Impacts of Salt-and-Pepper Noise

Besides Gaussian noise in wireless medium, there exists impulse noise, typically salt-and-pepper noise, which is caused by hardware failure in image detection. The impacts of salt-and-pepper noise are depicted in Figs. 15 and 16 where the PSNR and SSIM of the proposed PPC scheme are respectively plotted as functions of the salt-and-pepper noise density, i.e. d. The other simulation parameters are similarly set as in Fig. 13 for both VDSR and bicubic interpolation. It can be seen that the increase of noise density causes significant performance degradation.

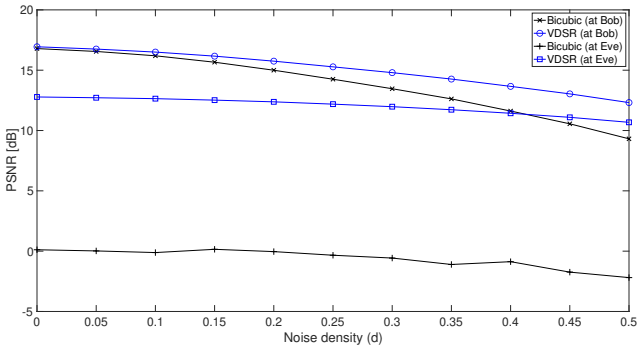

Fig. 15: PSNR versus salt-and-pepper noise density.

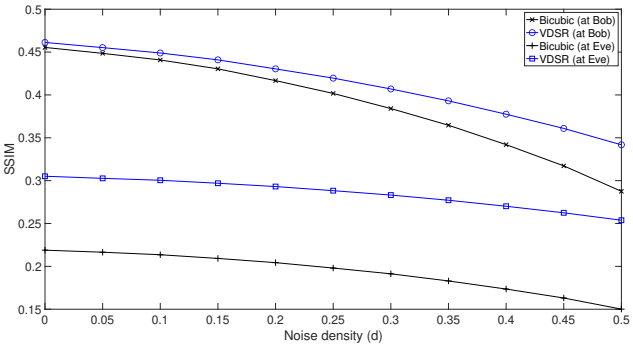

Fig. 16: SSIM versus salt-and-pepper noise density.

Specifically, the PSRN is reduced about 1 dB for every 10% increase in noise density. Also, the VDSR is shown to outperform the bicubic interpolation, especially when the noise density is high. For instance, with VDSR and $d = 0.5$, Bob can achieve 3 dB higher in PSNR than bicubic interpolation. As shown in Figs. 15 and 16, much higher PSNR and SSIM are achieved at Bob compared to those at Eve.

F. Impacts of Scaling Factor

The impacts of scaling factor are shown in Fig. 17 where the PSNR at Bob and Eve of the proposed PPC scheme are plotted against noise variance in AWGN model, i.e. σ^2, with different scaling factors, i.e. $\delta = 4$ and $\delta = 10$. VDSR is considered with the same settings as in Fig. 13. It can be seen in Fig. 17, the PSNR at Bob decreases considerably in the noiseless environment as the scale increases, while there is not much difference in the PSNR in the noisy environment. For instance, 4 dB is reduced when $\sigma^2 = 0$, while less than 1 dB when $\sigma^2 > 0.1$. This means

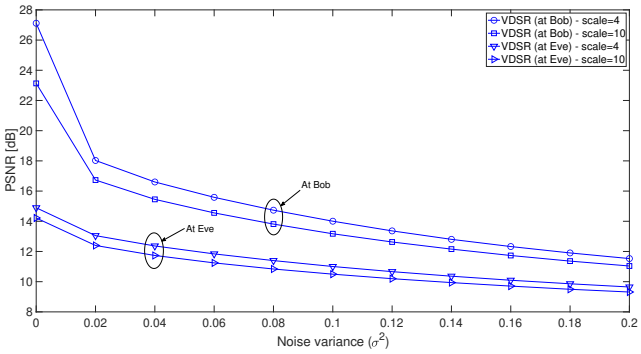

Fig. 17: PSNR versus noise variance with respect to different scaling factors.

that over the noisy medium, the image can be downscaled with a higher scaling factor to save the bandwidth with negligible performance loss.

VIII. Conclusions

In this paper, we have proposed a Deep-NC protocol for secure ISR communication between Alice and Bob taking into account two noise models including Gaussian noise and non-Gaussian or impulse noise in the form of salt-and-pepper noise. Specifically, two RLNC based schemes, i.e. PIC and PPC, have been devised to protect the secret image from Eve. It has been shown that Bob achieves a much higher PSNR than Eve up to 32 dB with the proposed Deep-NC protocol. The PPC, although outperforms the PIC scheme, requires a higher computation complexity to process all image pixels. Additionally, an enhanced performance has been shown to achieve at Bob in both noise models over the whole range of noise variance in Gaussian model and salt-and-pepper noise density. Furthermore, the Deep-NC protocol enables the original image to be downscaled to a much lower resolution prior to transmitting over the lossy environment, which accordingly implies the effectiveness of the proposed scheme in saving the transmission bandwidth. The future work would be the investigation of different image processing techniques for downscaling and upscaling incorporating in the Deep-NC protocol.

References

[1] Z. Wang, J. Chen, and S. C. H. Hoi, "Deep learning for image super-resolution: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–1, 2020.

[2] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[3] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 782–795, Oct. 2003.

[4] S. Zhang, S. C. Liew, and P. P. Lam, "Hot topic: Physical-layer network coding," in *Proc. ACM MobiCom'06*, Los Angeles, CA, USA, Sep. 2006, pp. 358–365.

[5] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the air: Practical wireless network coding," *IEEE/ACM Trans. Netw.*, vol. 16, no. 3, pp. 497–510, Jun. 2008.

[6] E. Soljanin, "Network multicast with network coding [lecture notes]," *IEEE Signal Process. Mag.*, vol. 25, no. 5, pp. 109–112, Sep. 2008.

[7] D. Nguyen, T. Tran, T. Nguyen, and B. Bose, "Wireless broadcast using network coding," *IEEE Trans. Veh. Technol.*, vol. 58, no. 2, pp. 914–925, Feb. 2009.

[8] E. Magli, M. Wang, P. Frossard, and A. Markopoulou, "Network coding meets multimedia: A review," *IEEE Trans. Multimedia*, vol. 15, no. 5, pp. 1195–1212, 2013.

[9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[10] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 6, pp. 1153–1160, 1981.

[11] P. Smith, "Bilinear interpolation of digital images," *Ultramicroscopy*, vol. 6, no. 1, pp. 201 – 204, 1981. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0304399181801994

[12] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 184–199.

[13] W. Lai, J. Huang, N. Ahuja, and M. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5835–5843.

[14] M. Irani and S. Peleg, "Improving resolution by image registration," *CVGIP: Graphical Models and Image Processing*, vol. 53, no. 3, pp. 231 – 239, 1991. [Online]. Available: http://www.sciencedirect.com/science/article/pii/104996529190045L

[15] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep back-projection networks for super-resolution," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1664–1673.

[16] Z. Li, J. Yang, Z. Liu, X. Yang, G. Jeon, and W. Wu, "Feedback network for image super-resolution," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3862–3871.

[17] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1646–1654.

[18] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, 2017.

[19] Y. Zhao, Y. Li, X. Dong, and B. Yang, "Low-frequency noise suppression method based on improved dncnn in desert seismic data," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 5, pp. 811–815, 2019.

[20] T. Rahim, S. Khan, M. A. Usman, and S. Y. Shin, "Exploiting de-noising convolutional neural networks dncnns for an efficient watermarking scheme: a case for information retrieval," *IETE Technical Review*, pp. 1–11, 2020.

[21] F. Guerrini, M. Dalai, and R. Leonardi, "Minimal information exchange for secure image hash-based geometric transformations estimation," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3482–3496, 2020.

[22] H. Peng, B. Yang, L. Li, and Y. Yang, "Secure and traceable image transmission scheme based on semitensor product compressed sensing in telemedicine system," *IEEE Internet Things J.*, vol. 7, no. 3, pp. 2432–2451, 2020.

[23] X.-Z. Xie, C.-C. Chang, and C.-C. Lin, "Reversibility-oriented secret image sharing mechanism with steganography and authentication based on code division multiplexing," *IET Image Processing*, vol. 13, no. 9, pp. 1411–1420, 2019.

[24] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," *IBM Systems Journal*, vol. 35, no. 3.4, pp. 313–336, 1996.

[25] E. Franz, A. Jerichow, S. Möller, A. Pfitzmann, and I. Stierand, "Computer based steganography: How it works and why therefore any restrictions on cryptography are nonsense, at best," in *Information Hiding*, R. Anderson, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 7–21.

[26] A. I. Hashad, A. S. Madani, and A. E. M. A. Wahdan, "A robust steganography technique using discrete cosine transform insertion," in *2005 International Conference on Information and Communication Technology*, 2005, pp. 255–264.

[27] P.-Y. Chen and H.-J. Lin, "A DWT based approach for image steganography," *International Journal of Applied Science and Engineering*, pp. 275–290, 2006.

[28] R. H. Chan, Chung-Wa Ho, and M. Nikolova, "Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization," *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1479–1485, 2005.

[29] C. E. Duchon, "Lanczos Filtering in One and Two Dimensions," *Journal of Applied Meteorology*, vol. 18, no. 8, pp. 1016–1022, 08 1979. [Online]. Available: https://doi.org/10.1175/1520-0450(1979)018¡1016:LFIOAT¿2.0.CO;2

[30] Y. Li, D. Liu, H. Li, L. Li, Z. Li, and F. Wu, "Learning a convolutional neural network for image compact-resolution," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1092–1107, 2019.

[31] M. Grubinger, P. D. Clough, H. Müller, and T. Deselaers, "The IAPR TC-12 benchmark: A new evaluation resource for visual information systems," 2006.

[32] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.

[33] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2472–2481.