CSIE-M: Compressive Sensing Image Enhancement Using Multiple Reconstructed Signals for Internet of Things Surveillance Systems

Chi Do-Kim Pham [iD], Jian Yang [iD], and Jinjia Zhou [iD], *Member, IEEE*

***Abstract*—Artificial intelligence of things has brought artificial intelligence to the cutting-edge Internet of Things. In recent years, compressive sensing (CS), which relies on sparsity, is widely embedded and expected to bring more energy efficiency and a longer battery lifetime to IoT devices. Different from the other image compression standards, CS can get various reconstructed images by applying different reconstruction algorithms on coded data. Using this property, it is the first time to propose a deep learning based compressive sensing image enhancement framework using multiple reconstructed signals (CSIE-M). In this article, first, images are reconstructed by different CS reconstruction algorithms. Second, reconstructed images are assessed and sorted by a no-reference quality assessment module before being input to the quality enhancement module by order of quality scores. Finally, a multiple-input recurrent dense residual network is designed for exploiting and enriching the useful information from the reconstructed images. Experimental results show that CSIE-M obtains 1.88–8.07 dB peek-signal-to-noise (PSNR) improvement while the state-of-the-art works achieve a 1.69–6.69 dB PSNR improvement under sampling rates from 0.125 to 0.75. On the other hand, using multiple reconstructed versions of the signal can improve 0.19–0.23 dB PSNR, and only 4% reconstructing time is increasing compared to using a reconstructed signal.**

***Index Terms*—Compressive sensing (CS), deep learning approach for compressed image enhancement, multiple-to-one mapping.**

I. INTRODUCTION

INTERNET of Things (IoT) interconnects numerous devices including sensors, cameras, smart home products, and smart city products in an environment. It provides a fundamental way to communicate, store, transmit, and process sensed data, giving better productivity and more efficient solutions for improving the quality of human life. Surveillance systems have been pointed out as one of the most necessary but challenging solutions in urban developments due to large data storage requirements and the high computational complexity in processing images and videos sensed by cameras. Therefore, compression methods that can adapt the requirements of 1) saving the power consumption and prolonging the battery lifetime of IoT devices, 2) securing the data, and 3) balancing the traffic load when traveling throughout the network are preferred in designing sensing devices for surveillance systems. Traditional Shannon–Nyquist theorem states that signals need to be sampled at twice the bandwidth to be recoverable. In IoT systems such as remote surveillance and astronomy satellites, the Shannon–Nyquist rate is costly, requires ample storage space, and wide bandwidth for transmission. Compressive sensing (CS), which requires only a few compressive measurements to contain nearly all the useful information, breaks the limitation stated by Shannon–Nyquist's theory. CS has adapted the requirements and becomes one of the effective lossy compression methods that are considered when designing devices for IoT applications [1]–[3]. First, in the CS encoder, only matrices multiplication is performed. The matrices multiplication is simple to be embedded in resource-limited devices and ensures energy saving for IoT self-powered devices. Second, the measurement matrix is only shared between the encoder and decoder, making the network secure. Third, the amount of sent measurement is fixed throughout the time, ensuring the traffic for transmission. Unlike the other compression standards, reconstructing CS signals is to solve a nonlinear inverse problem [4]. Since different CS reconstruction algorithms dissimilarly model the recovery solution, choosing a CS reconstruction algorithm for an application is challenging. On the other hand, there is always space to further improve the signals reconstructed by lossy compression methods. In this article, a study on enhancing CS reconstructed images is
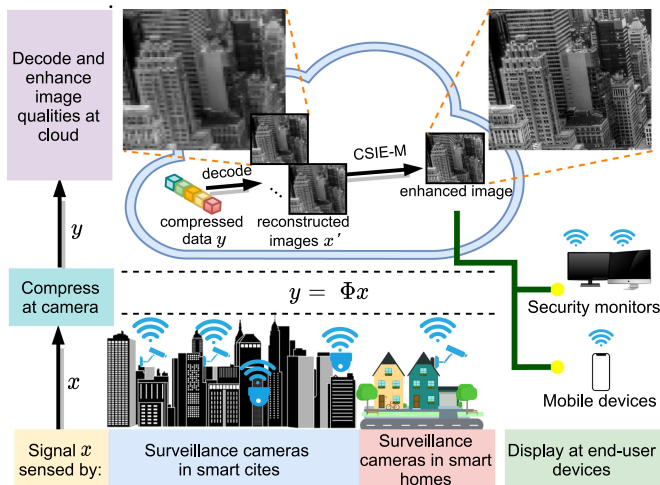
Fig. 1.   Proposed CSIE-M in IoT surveillance system.

proposed to improve the qualities of CS reconstructed signals without changing the encoder and the decoder, keeping the requirements of IoT solutions.

Convolutional neural network (CNN) is well-known for learning complex functions where the designed filters are not flexible enough to model. In recent years, CNN has been widely applied and has obtained remarkable results in image quality enhancement. In this work, we take the next step toward an artificial intelligence of things (AIoT) approach for enhancing CS reconstructed images sensed by IoT devices (Fig. 1). Images are sensed and encoded in IoT cameras before being sent to the IoT cloud. Unlike the existing image enhancement works, where enhancing the image relies on a single degraded image [5]–[11] or neighboring frames [12], [13], this work takes advantage of different CS reconstructed images with different qualities and introduces a deep learning based compressive sensing image enhancement framework using multiple reconstructed signals (CSIE-M). At IoT cloud, compressed data are decoded by the commonly used CS reconstruction algorithms: L1 equality constraints via primal-dual algorithm (L1EQPD) [14], spectral projected-gradient for L1 (SPGL1) [15], orthogonal matching pursuit (OMP) [16], and sparsity adaptive matching pursuit (SAMP) [17]. In learning multiple-to-one mapping, it is necessary to decide which branch an image should be input. Therefore, a no-reference quality assessment module, namely Scorenet, is proposed to score and rank reconstructed images before feeding them to the quality enhancement network by order of quality. In the quality enhancement module, a multiple-input residual recurrent network (MRRN) is proposed for enhancing the reconstructed images by exploiting and enriching useful features via a recurrent mechanism. MRRN takes the best quality image, which will be added to the enhanced feature for the main branch. The two lower quality images are input to the supporting branches. Finally, enhanced images at the IoT cloud are displayed in monitoring devices for surveillance solutions or transfer to other image processing tasks such as recognition and

detection. The experimental result shows that CSIE-M improves 1.88–8.07 dB peek-signal-to-noise (PSNR) compared to the main input image on various sampling rates (SRs) from 0.125 to 0.75.

This work's main contributions are summarized as follows. First, a CSIE-M is designed for the first time. In CSIE-M, a no-reference quality assessment module scores and ranks reconstructed images before feeding them to the quality enhancement module. Second, the proposal outperforms the state-of-the-art works in distorted image enhancement. Finally, a study on the number of input images is also conducted to show the effect of using multiple inputs on enhancing the reconstructed image quality.

II. RELATED KNOWLEDGE

A. Compressive Sensing

Given sensed signal x that can be represented by a $n \times 1$ sparse vector s in the domain ψ, CS encoder simply calculates the $m \times 1$ measurement y by

$$y = \Phi x = \Phi \psi s \tag{1}$$

where Φ is the $m \times n$ measurement matrix fixed in the encoder and the decoder [4]. The commonly designed measurement matrices include random matrices, binary matrices, and structural matrices. The quotient of m and n defining the system's compression ratio, also known as sampling rate, represents the amount of data sent to the decoder. The process of decoder, on the other hand, is more complicated. In the CS-based image compression, the decoder reconstructs the image x back into the pixel domain by solving an underdetermined matrix equation where $m < n$. Therefore, reconstructing x at the decoder is solving an ill-posed problem. There have been many CS reconstructing algorithms including greedy algorithms, convex optimization, and gradient-based algorithms. The proposal adopts L1 optimization that include L1EQPD [14] and SPGL1 [15], and greedy algorithms include OMP [16] and SAMP [17] for CS image reconstruction.

B. Deep Learning Based Distorted Image Enhancement

Image enhancement is one of the essential components in image processing and image-display applications [18], [19]. Concerning deep learning based distorted image enhancement, single image enhancement [5]–[11], and the multiframe enhancement, [12] and [13] are mainly focused in removing compression artifacts and denoising. Zhang et al. [5] introduce a denoising convolutional neural network (DnCNN) that can deal with different Gaussian noise levels, single image super-resolution, and JPEG image artifacts caused by different quality factors. For denoising real-noisy images, Anwar and Barnes [7] introduce a novel single-stage blind real image denoising network (RIDNet). In RIDNet, local skip connections, short skip connections, and long skip connections are utilized to exploit low-frequency information over the feed-forward. Jia et al. [6] introduce a content-aware loop filtering scheme based on
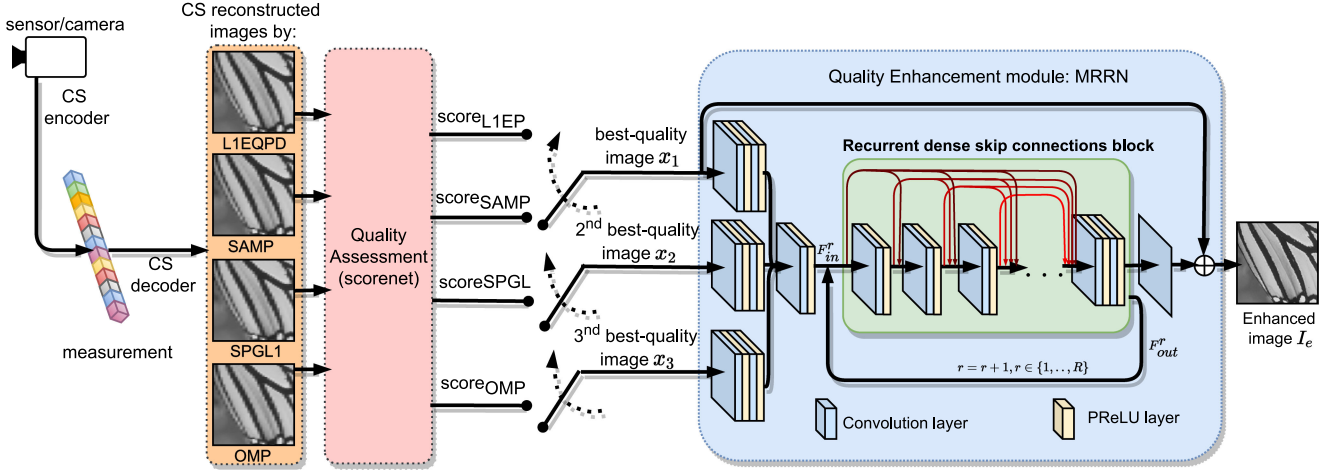
Fig. 2. Proposed CSIE-M architecture. Sensed signal is compressed by CS in cameras. At the decoder, compressed data (measurement) is recovered by different CS reconstruction algorithms. Reconstructed images are judged by the no-reference quality assessment module Scorenet and fed to the quality enhancement network MRRN by order of quality scores for producing the enhanced image.

multiple CNN models (CACNN) for improving the performance of the high efficiency video coding (HEVC) by enhancing the quality of decoded frames. In [8], the authors introduce attention-guided denoising convolutional neural network (ADNet). AD-Net combines dilated convolutions, standard convolutions, and an attention mechanism for real-noisy image denoising and blind denoising. In [9], both noise removal and noise generation tasks are trained in a Bayesian network that learns the joint distribution of the pairs of the clean and distorted images. The authors in [10] propose a block artifact removing convolutional neural networks (BARCNN) for JPEG image enhancement. BARCNN can be integrated to the receiver side to enhance image quality without any additional cost on the IoT node ends. Building upon DnCNN, the authors in [11] propose a theoretically grounded blind and universal deep learning image denoiser, namely blind universal image fusion denoiser (BUIFD), for additive Gaussian noise removal. Recent approaches [12], [13] take advantage of the temporal correlation between adjacent frames to enhance the low-quality image by using its neighboring high-quality video frames. Apart from these works, this work proposes a deep network that enriches and synthesizes useful information via a recurrent mechanism performing on extracted features of CS decoded images.

III. PROPOSED CSIE-M FRAMEWORK

A. Overview of the CSIE-M Framework

As mentioned above, there have been many approaches for solving the nonlinear inverse problem at CS decoder. Different CS reconstruction methods model the solution differently, resulting in images being recovered with different qualities. Using different CS reconstructed images provides more representations for CNN to exploit and enhance the performance in recovering the original signals. To make full use of this property, we propose a deep learning based CSIE-M that learns a multiple-to-one mapping from the reconstructed images to the original one (shown in Fig. 2).

First, a sensed signal is encoded by CS in sensors or cameras. At the decoder, different CS reconstruction algorithms perform on the compressed data to obtain different reconstructed images. Let $X = \{x_{\text{alg}}, \text{alg} \in \{\text{L1EQPD}, \text{SPGL}, \text{OMP}, \text{SAMP}\}\}$ indicate the images reconstructed by the four algorithms L1EQPD, SPGL, OMP, and SAMP. In CSIE-M, the highest quality reconstructed image takes the highest responsibility in generating the enhanced image. The other inputs are considered additional features generated by designed filters: CS reconstruction algorithms. We propose a no-reference quality ranking module including a deep learning based no-reference image quality assessment (IQA) Scorenet for scoring and ranking CS reconstructed images. Scorenet predicts the quality score z_{alg} of image x_{alg} as

$$z_{\text{alg}} = f_{\text{scorenet}}(x_{\text{alg}}). \tag{2}$$

Images in list X are sorted based on the corresponding quality scores in list Z. The best quality image x_1, the second best quality image x_2, and the third best quality image x_3 are fed to MRRN denoted as f_{MRRN}, and the enhanced image I_e can be formulated as

$$I_e = f_{\text{MRRN}}(x_1, x_2, x_3). \tag{3}$$

B. No-Reference Quality Assessment Module: Scorenet

In reality, the original image does not always exist. Therefore, full-reference metrics PSNR and structural similarity index (SSIM) cannot be used for assessing the distorted image quality. In this work, we propose a deep learning based no-reference quality assessment module, called Scorenet, to guide the enhancement module. Our Scorenet simulates full-reference IQA metrics: estimating the difference between the distorted image and the reference image. Scorenet (shown in Fig. 3) includes two main components: the reference generative net G for generating the pseudo-original image and the quality-score prediction network S predicting the quality score of the distorted image. Given the distorted image x, our objective is to infer the
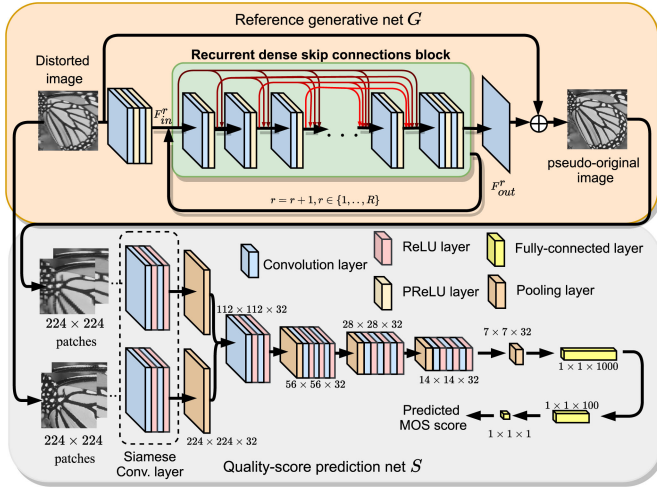
Fig. 3. Proposed Scorenet architecture. A distorted image is input to the reference generative net G for the corresponding pseudo-original image. Distorted image and the pseudo-original image are then split into 224×224 patches by a stride of 40 before being fed to quality-score prediction net S. The final predicted score of the input image is the average score of all patches.

quality score z. In full-reference IQA tasks, quality scores can be calculated by comparing the distorted image and the reference image. Scorenet simulates the full-reference IQA by estimating the pseudo-original image $y' = f_{\theta_G}(x)$ where $f_{\theta_G}(x)$ indicates the learnt mapping of reference generative net G. Predicting score z of an image x can be formulated as

$$z' = f_{\text{scorenet}}(x) = f_{\theta_S}(x, f_{\theta_G}(x)) \qquad (4)$$

where θ_G and θ_S denote the learnt parameters of G and S, respectively. In general, the reference generative net G has the same purpose of the quality enhancement module MRRN: to generate undistorted images. We design the reference generative net and the quality enhancement module to share some architectures and the loss function. The difference between the reference generative net architecture and MRRN architecture is the number of inputs, where the prior takes one, the latter takes three.

We define the basic convolution layer in our network as $\text{Conv}(k, s)$ with k kernels size $s \times s$. For the reference generative net G, the distorted image is first fed into two convolution groups, and each group is defined as $\text{Conv}(32, 3) \to ParametricRectifiedLinearUnit(PReLU) \to \text{Conv}(32, 3) \to \text{PReLU}$. In reference generative net G, PReLU activation function follows all the convolution layers except the final one. We set stride and padding to one during convolution. The main part of the reference generative net G is the recurrent dense skip connection block (explained in Section III-C), also used in MRRN. Feature maps output from the recurrent dense skip connection block are synthesized in the final $\text{Conv}(32, 3)$ producing an enhanced feature. This enhanced feature is added onto the distorted image x for a pseudo-original image y'.

In the quality-score prediction net S, rectified linear unit (ReLU) activation function follows all the convolution layer

$\text{Conv}(32, 3)$. Our network, inspired by the VGG16 model [20], aims to infer the quality score, given the pair of a distorted image and the corresponding pseudo-original image. Image is first split into 224×224 patches with a stride of 40. A pair of 224×224 patches from the distorted image x and pseudo-original image y' are fed to a group of siamese convolution layer that includes $\text{Conv}(32, 3) \to \text{ReLU} \to \text{Conv}(32, 3) \to \text{ReLU}$. We use a siamese convolution on both the distorted and pseudo-original images to extract comparable feature maps. The rest of the net then focuses on finding the difference between these feature maps. Feature maps after siamese convolutional layer will be pooled by a pooling layer with a window of 2×2 to the size of 112×112. Pooled feature maps from two inputs will be concatenated before feeding into the network. The output from the final fully connected layer is the predicted mean opinion score (MOS) score of a patch. The score of the entire image is the average score of all patches. $L2$ has been chosen as the loss function. Training S on N training samples becomes minimizing loss function L_S

$$L_S = \frac{1}{2N} \sum_{i=1}^{N} (z_i - f_{\theta_S}(x_i, y'_i))^2. \qquad (5)$$

C. Quality Enhancement Component: Multiple-Input Residual Recurrent Network

Recently, Li *et al.* [21] introduced an image super-resolution feedback network (SRFBN) which has achieved outstanding results in image super-resolution tasks. Inspired by [21], this work introduces a feedback mechanism for enhancing the quality of input images. Our network architecture MRRN (Fig. 2) includes the main branch, two supporting branches, a recurrent dense skip connections block (RDBlock), and a global skip connection for residual learning. x_1, x_2, and x_3, respectively, denote the best, the second best, and the third best quality images by order of the predicted MOS scores. The original image before CS encoding is denoted as y. The goal of our quality enhancement module is to learn the mapping f_{MRRN} between the input images x_1, x_2, and x_3 and the target enhanced image y'

$$y' = f_{\text{MRRN}}(x_1, x_2, x_3; \Theta). \qquad (6)$$

Input images are fed into a group of $\text{Conv}(n_f, 3) \to \text{PReLU} \to \text{Conv}(n_f, 3) \to \text{PReLU}$ in each branch. In the quality-score prediction net S, the first convolution group extracts the same features from the distorted image and the pseudo-original image. The rest of the network S aims to find the differences between these features. MRRN, on the other hand, aims to keep the diversity of extracted features from different input images. Therefore, convolution layers for each input are preferred over a siamese convolution layer. We formulate the output feature map f at convolution layer lth at branch k as

$$f_l^k = PReLU(f_{l-1}^k * w_l^k + b_l^k) \qquad (7)$$

where w_l^k and b_l^k are, respectively, the learnt weight and bias of convolution layer lth in branch k and f_0^k is $\{x_k, k = 1, 2, 3\}$. After the first two convolution layers, n_f feature maps in each

branch will be concatenated to a convolution layer f_3

$$f_3 = \text{PReLU}(\text{cat}(f_2^1, f_2^2, f_2^3) * w_3 + b_3) \qquad (8)$$

where cat $(f_1, f_2, \ldots, f_n)$ denotes the concatenate operation for the list of the feature maps $f_1, f_2, \ldots, f_n$ on channel dimension. Later, the output from the fourth convolution layer f_4 is input to the RDBlock. Stride for every convolution layer is set as one. To keep the size of input image through doing convolution, all the paddings are set as $(s-1)/2$.

Feedback mechanism. We introduce the RDBlock that integrates the two elements: depth and skip connections. In RDBlock, no information is omitted: The extracted low-level features F_{in}^r are reused in each loop and inside the RDBlock, and the high-level features F_{out}^{r-1} are also used to fine-tune the low-level feature input F_{in}^r of RDBlock. RDBlock does recurrence for R times; each $r = \{1, 2, \ldots, R\}$ corresponds to an output $\hat{y}_r$ of the network. In the loop rth, RDBlock returns an output of F_{out}^r given a pair of F_{in}^r and F_{out}^{r-1}

$$F_{\text{out}}^r = \text{RDBlock}(F_{\text{in}}^r, F_{\text{out}}^{r-1}). \qquad (9)$$

F_{out}^{r-1} at $r = 1$ is set as F_{in}^r. For $r \geq 2$, the output feature maps F_{out}^r are stored and be concatenated with the F_{in}^r for the next loop until $r = R$. In training recurrent neural network, the feedback mechanism receives the information from previous loop $(r-1)$th to further improve the input of the recurrent mechanism. We denote g_0 as the input of the RDBlock

$$g_0 = \text{cat}(F_{\text{in}}^r, F_{\text{out}}^{r-1}). \qquad (10)$$

Our RDBlock is built from nine convolution layers and dense skip connections. In each loop rth, the input g_0 and feature maps from the lower level layers are concatenated and fed to the higher level ones. Let $g_{j,j \in [1,9]}$ denote output from convolution layer jth. In our RDBlock, all layers jth, excluding the final one, synthesize information from $(j-1)$ previous layers and the input g_0. The final layer g_9 takes only feature maps output by convolution layer g_8. The output of each convolution layer at layer jth, $j < 9$, in the RDBlock is defined as

$$g_j = \text{PReLU}(\text{cat}(g_0, g_1, \ldots, g_{j-1}) * w_j + b_j). \qquad (11)$$

Convolution layer jth with $j \in [1, 8]$ in RDBlock takes an input of $(j+1) \times n_f$ concatenated feature maps and outputs n_f feature maps. In this work, we set the value of n_f to 32 for all convolution layers in RDBlock, and the number of loop R is set as four. There is no PReLU activation for the final convolution of MRRN. The enhanced feature from each loop will be added onto the best quality image input x_1 for the output $\hat{y}_r$. There is only one ground-truth label y_0 for R outputs of the network. In testing, only output from the final loop y_R is considered the result of the network. However, all the outputs $\hat{y}_r$ are used to calculate the loss function

$$L(\Theta) = \frac{1}{M.R} \sum_{i=1}^{M} \sum_{r=1}^{R} I^r \parallel y_0^i - \hat{y}_r^i \parallel_1 \qquad (12)$$

where M is the number of training samples, and Θ denotes the learned network parameters. L1 loss is used to optimize the

Metric	PSNR	SSIM	Scorenet (Ours)
SROCC	0.571	0.604	0.731
LCC	0.593	0.635	0.774

Note: Blue indicates the best result.

network parameters. I^r is the importance of output r in the outputs list. Following [21], we set I^r to 1 for all the outputs.

IV. EXPERIMENTAL RESULTS AND COMPARISON

A. Experiment Settings

MRRN settings. For training MRRN, we use images with different sizes from DIV2K [22] for the training set and images from BSD500 training set [23] for evaluating during training. Set 5 [24], Set 14 [25], Urban100 [26], and 200 images from the testing set of BSD500 [23] are used for testing. The inverse fast Walsh–Hadamard transform and binary Hadamard matrix are applied for transforming and measuring the input images. The other CS reconstruction algorithms and other matrices can replace the reconstruction algorithms and measurement matrix used in CSIE-M. Images in the training and testing sets are fed to the Scorenet to get the ranking scores and then fed to the quality enhancement module MRRN by order of quality scores.

Scorenet settings. For training Scorenet, we mostly focus on the dataset TID2013 [27], which is commonly used for learning no-reference IQA tasks. TID2013 contains a total of 3000 images generated by 24 types of distortions. We randomly divide reference images into 80% for training and 20% for testing as the no-reference IQA works have done [28]. Distorted images will go to the set that its reference image belongs to, ensuring no overlap between the training and testing sets. In testing, 20% images in the testing set are used for evaluating our Scorenet. Like the other IQA works [28], we evaluate the efficiency of our Scorenet via Spearman's rank order correlation coefficient (SROCC) and the linear correlation coefficient (LCC). The higher SROCC and LCC represent the higher correlation between the predicted scores and the human-ranked MOS scores.

Training settings. Our experiments on the CSIE-M framework are conducted on Pytorch 1.0.0 with NVIDIA Tesla V100 GPUs' support. Adam optimization is used in training all the networks in CSIE-M. In training MRRN, the learning rate starts at 0.0001 and is divided by 2 every 300 epochs. The commonly used PSNR and SSIM metrics are applied for evaluating quality enhancement results. For these metrics, the higher value indicates a better result. Four models for four SR of 0.125, 0.25, 0.5, and 0.75 are trained and converged in different training epochs.

B. Ablation Studies

Study on Scorenet. Table I presents the ten-time average of SROCC and LCC metrics of our Scorenet compared to the widely used full-reference IQA metrics PSNR and SSIM. Observe from Table I that our Scorenet performs closer prediction to

Number of MRRN inputs	Max quality diff. between inputs ΔPSNR / ΔSSIM	Random-based input ranking PSNR / SSIM	Scorenet-based input ranking PSNR / SSIM
1	0.23 / 0.21	31.647 / 94.862	32.772 / 95.207
2	0.218 / 0.21	31.747 / 94.882	32.964 / 95.36
3	0.33 / 0.22	31.882 / 94.94	33 / 95.387
4	0.33 / 0.21	31.871 / 94.946	32.978 / 95.384

the human eyes (presented in MOS score) than the full-reference IQA metrics PSNR and SSIM. In detail, we obtain SROCC relative improvements of 28.02% and 21.02% compared to PSNR and SSIM metrics, respectively. In the LCC metric, improvements of 30.52% and 21.88% are achieved compared to PSNR and SSIM metrics. These results show an adequate ability of Scorenet to guide MRRN on enhancing the reconstructed image quality.

To verify the effectiveness of the proposed Scorenet, Scorenet has been replaced by a random-based input ranking. We perform a ten-time random-based input ranking and average these results for comparison. Table II shows the quality comparison of using random-based ranking and Scorenet-based ranking. As observed from Table II, applying Scorenet can gain up to 1.127 dB and 4.8×10^{-3} on average PSNR and SSIM improvements compared to randomly setting the input images of MRRN when it comes to three-input MRRN. On the other hand, CSIE-M with multiple-input images and Scorenet can significantly improve the reconstructed image quality compared to using one input image or randomly set the order for multiple input images.

Study on the number of input images. This experiment validates the effect of the number of input images on the performance of enhancing CS reconstructed images. We train our dataset on single-input, two-input, three-input, and four-input MRRN. The N-input MRRN takes N images ranked by the no-reference quality ranking module. To fully evaluate the performance of N-input MRRN, reconstructing time is also considered besides PSNR and SSIM. Table II shows the PSNR and SSIM of N-input network, $N \in \{1, 2, 3, 4\}$. We choose an SR of 0.5 for validating this experiment. Obtain from Table II, the three-input MRRN obtains the highest performance compared to the other N-input MRRNs. From single-input MRRN to two-input MRRN, average PSNR and SSIM have significantly increased by 0.19 dB and 0.002, respectively. It is well-known in learning deep networks that more the representations that are obtained, the better the results that can be achieved. In learning to enhance CS reconstructed images, other supporting images x_2 and x_3 can be considered the representations obtained from special filters: CS reconstruction algorithms. Moreover, the reconstructing time increases the amount of 5 ms from single-input MRRN to two-input MRRN. That has shown the advantage of multiple-input MRRN compared to single-input MRRN: The image quality is significantly improved, and the running time slightly increases. We report the small differences in performance of multiple-input CSIE-M networks, and the best

performance, over the test images, belongs to the three-input MRRN.

Performance on other CS reconstruction algorithms. We present the CSIE-M performance on different CS reconstruction algorithms in Table III. In this experiment, four other CS reconstruction algorithms from convex optimization algorithms that include two-step iterative shrinkage/thresholding (TwIST) [29] and group-sparse basis pursuit (GroupBP) [30], and from greedy algorithms that include compressive sampling matching pursuit (CoSaMP) [31] and stagewise orthogonal matching pursuit (StOMP) [32] have been used. Over the testing sets, quality difference between images reconstructed by the above algorithms is up to 1.5, 2.07, 3.22, and 2.79 dB PSNR, while the CS reconstruction algorithms used in the original proposal is up to 0.04, 0.07, 0.33, and 0.2 dB PSNR at SRs of 0.125, 0.25, 0.5, and 0.75, respectively.

Table III illustrates the quantitative results (PSNR and SSIM) comparison among the combinations of CS reconstruction algorithms: the proposed CSIE-M, the combination of the other four CS reconstruction algorithms (CSIE-M*), the combination of the greedy algorithms (CSIE-M**), the combination of convex optimization algorithms (CSIE-M***), the other combinations of CSIE-M used greedy algorithms and other convex optimization algorithms (CSIE-M****), the combination of CSIE-M used convex optimization algorithms and the other greedy algorithms (CSIE-M*****). It can be seen that CS reconstruction algorithms used in CSIE-M obtain the highest quality in terms of PSNR and SSIM over all the SRs. Generally, the other combinations reduce the quality of the enhanced images up to an amount of 0.136 dB PSNR and 3.8×10^{-2} SSIM on average. Also, note that even though two CS reconstruction algorithms are shared between some groups, the results are different since the input images fed to the network are different. For example, OMP, SAMP, and L1EQPD reconstructed images of *foreman* (Set14) have been fed into MRRN as the best quality, the second best quality, and the third best quality images, respectively. Meanwhile, the feeding order in CSIE-M** is, respectively, StOMP, OMP, and SAMP reconstructed images. It concludes that choosing the top-three CS reconstruction algorithms and the order of input images of MRRN takes a high responsibility on generating high-quality images.

Study on numbers of iteration R of the recurrent dense skip connections block. In this experiment, we find the correlation between R and network performance. The investigation is conducted on the three-input network under the SR of 0.5. We separately train five CSIE-M models in which the recurrent iteration R is set as 1, 2, 3, 4, and 5. In Table IV, a clear trend is performed: the larger the R is, the better the quality of the image. At $R = 1$, the network is considered a traditional neural network. It obtains the lowest image quality compared to using the recurrent neural network where $R \geq 2$. The reconstructing time between different R's is about 4 ms if the number of loops is increased by one. This experiment will consider the performance and training time of different R's. For $R = 1$, the network takes 45 h to converge. For recurrent network $R \geq 2$, it takes about three days for training the network whose recurrent iteration is

TABLE III
CSIE-M Performance in Terms of PSNR and SSIM Comparison on Difference Combinations on CS Reconstruction Algorithms

SR	CSIE-M		CSIE-M*		CSIE-M**		CSIE-M***		CSIE-M****		CSIE-M*****	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
0.125	26.824	0.825	26.815	0.813	26.811	0.786	26.823	0.814	26.781	0.813	26.823	0.814
0.25	27.747	0.864	27.612	0.849	27.691	0.849	27.665	0.849	27.707	0.853	27.746	0.853
0.5	33	0.953	32.924	0.949	32.951	0.948	32.905	0.949	32.982	0.949	32.981	0.949
0.75	38.24	0.984	38.193	0.982	38.186	0.982	38.198	0.982	38.201	0.982	38.164	0.982

Note: Blue indicates the best result and orange indicates the second best result. Reconstruction algorithms for inputs: CSIE-M: L1EQPD, SPGL, OMP, and SAMP. CSIE-M*: TwIST, StOMP, GroupBP, and CoSaMP. CSIE-M**: SAMP, OMP, CoSaMP, and StOMP. CSIE-M***: GroupBP, SPGL, L1EQPD, and TwIST. CSIE-M****: SAMP, OMP, TwIST, and GroupBP. CSIE-M*****: CoSaMP, L1EQPD, SPGL, and StOMP.

TABLE IV
Study on Numbers of the Recurrent Iteration R of the Network Under PSNR and SSIM $\times 10^{-2}$ at Sampling Rate 0.5

Dataset	R = 1		R = 2		R = 3		R = 4		R = 5	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Set 5	37.485	97.16	37.552	97.173	37.559	97.177	37.596	97.19	37.582	97.196
Set14	33.657	94.92	33.673	94.92	33.668	94.94	33.74	94.927	33.767	94.948
BSD500	33.278	95.429	33.318	95.452	33.371	95.493	33.403	95.509	33.421	95.525
Urban100	31.544	94.893	31.632	94.945	31.773	95.056	31.86	95.115	31.923	95.155
Average	32.817	95.266	32.871	95.297	32.949	95.358	33	95.387	33.032	95.41

Note: Averages are calculated over the number of images. Blue indicates the best result and orange indicates the second best result.

TABLE V
ΔPSNR and ΔSSIM$\times 10^{-2}$ Comparison to the State of the Arts

Dataset	SR	DnCNN TIP'17 [5]		RIDNet ICCV'19 [7]		CACNN TIP'19 [6]		ADNet N.N. '20 [8]		BUIFD TIP'20 [11]		CSIE-M (ours)	
		ΔPSNR	ΔSSIM	ΔPSNR	ΔSSIM	ΔPSNR	ΔSSIM	ΔPSNR	ΔSSIM	ΔPSNR	ΔSSIM	ΔPSNR	ΔSSIM
Set5	1/8	3.12	5.99	3.23	6.72	3.06	5.94	3.37	6.44	3.38	6.31	3.71	6.76
	1/4	2.81	4.25	3.02	4.96	2.61	4.09	3.01	4.91	3.02	4.59	3.46	5.14
	1/2	4.46	2.73	4.3	2.85	4.65	2.82	4.79	2.86	4.42	2.74	5.05	3.04
	3/4	5.84	2.54	5.86	2.86	7.31	2.99	5.8	2.77	7.11	2.9	8.59	3.28
Set14	1/8	2.05	5.21	2.08	5.72	2	5.26	2.14	5.75	2.14	5.5	2.32	5.96
	1/4	1.78	3.4	1.9	3.86	1.63	3.34	1.83	3.77	1.85	3.62	2.08	4.18
	1/2	3.27	3.52	3.18	3.66	3.37	3.58	3.45	3.67	3.13	3.49	3.62	3.82
	3/4	5.03	3.71	5.15	4.09	6.48	4.45	4.88	3.91	6.61	4.55	7.75	4.87
BSD500	1/8	1.43	4.58	1.45	4.93	1.41	4.64	1.5	4.97	1.51	4.93	1.65	5.3
	1/4	1.16	2.78	1.24	3.12	1.13	2.8	1.26	3.12	1.24	3.11	1.46	3.57
	1/2	2.92	3.76	2.81	3.95	3.03	3.8	3.15	4.14	2.91	3.77	3.4	4.15
	3/4	5.08	4.12	5.37	4.69	6.57	4.8	5.01	4.44	6.4	4.7	7.73	5.19
Urban100	1/8	2.06	6.62	2.1	7.37	2.07	6.88	2.24	7.73	2.14	7.18	2.6	8.4
	1/4	1.91	4.94	2.09	5.71	1.82	4.8	2.09	5.71	1.91	5.19	2.61	6.65
	1/2	3.88	5.37	3.59	5.37	4.04	5.52	4.32	5.98	3.57	5.15	4.86	6.16
	3/4	5.49	5.39	5.53	5.81	6.77	6.15	5.36	5.63	7.27	6.26	8.78	6.95
Average	1/8	1.68	5.27	1.71	5.76	1.67	5.39	1.79	5.89	1.77	5.68	2.01	6.32
	1/4	1.45	3.51	1.56	4	1.39	3.47	1.57	3.99	1.51	3.81	1.88	4.59
	1/2	3.26	4.24	3.09	4.36	3.38	4.31	3.56	4.67	3.15	4.18	3.89	4.75
	3/4	5.21	4.48	5.42	4.99	6.64	5.18	5.13	4.76	6.69	5.16	8.07	5.7

Note: Blue indicates the best result and orange indicates the second best result.

2, 3, and 4. For $R = 5$, it takes more than three days to converge on the DIV2K dataset. Observing from Table IV, the network with $R = 5$ shows the best performance; however, on the other hand, the training time increases. The recurrent iteration of four has average performance while the complexity is in the middle. The following experiments take a feedback iteration of four for analysis.

C. Overall Results

Table V shows our results in ΔPSNR and ΔSSIM$\times 10^{-2}$ of our proposal CSIE-M compared to the main input x_1 since x_1 takes the highest responsibility in generating the enhanced image. Generally, CSIE-M strongly improves the quality of CS reconstructed images. In the PSNR evaluation metric, the best and the lowest improvements belong to the SR of 0.75 and 0.25, respectively. In other words, CSIE-M improves an average PSNR of 2.01, 1.88, 3.89, and 8.07 dB for the SRs of 0.125, 0.25, 0.5, and 0.75, respectively. Under the SSIM evaluation, the best and the lowest improvements belong to the SR of 0.125 and 0.25, respectively. Notably, CSIE-M shows outstanding results on dataset Urban100 which contains aliasing edges and complex scene structures. We obtain improvements of 4.71 dB PSNR and 7.04×10^{-2} SSIM on average over all the SRs on Urban100.

Comparison to the state of the art: For a fair comparison, we retrain the related work models DnCNN [5], RIDNet [7], CACNN [6], ADNet [8], and BUIFD [11] on our training set.
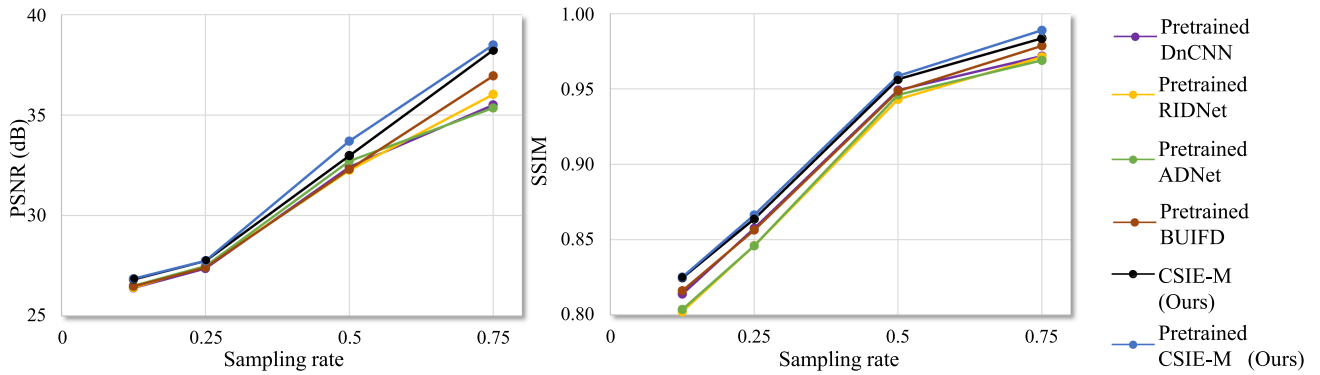
Fig. 4. PSNR and SSIM comparison to the pretrained state of the art in distorted image enhancement under sampling rates of 0.125–0.75.

For the image denoising works [5], [7], [8], [11], we set the CS reconstructed images as the noisy input image, and the original image is set as the clean ground truth. For enhancing the decoded image by HEVC [6], we assign the network to take an input and a ground truth of reconstructed images and the corresponding original image, respectively. All the settings are default. No noise or preprocessing is added to the training data of these related works. Since MRRN and related works' inputs are from different CS reconstruction algorithms, it is better to compare how much quality improvements each approach can obtain. In this comparison experiment, we assume the highest quality image in PSNR for the related works. While the difference in image quality of the related works' input and CSIE-M main input x_1 is 0.02 dB PSNR, the proposal significantly improves the image quality in PSNR and SSIM. In other words, the related works increase from 1.39 to 6.69 dB, where ours is from 1.88 to 8.07 dB in average PSNR improvement over the testing images. In terms of SSIM, our improvement is 4.59×10^{-2} to 6.32×10^{-2} while that of others are from 1.45×10^{-2} to 5.68×10^{-2} on average. Our proposal shows significant improvements compared to the related works in a high SR of 0.75 and complex structures such as edge and aliasing in the test set Urban100. At an SR of 0.75, CSIE-M scores an improvement of 8.07 dB and 5.7×10^{-2} in terms of PSNR and SSIM, where the related works are up to 6.69 dB PSNR and 5.16×10^{-2} SSIM improvement in average.

We also make a comparison to the pretrained related works (Fig. 4). In this experiment, we compare our models using training from scratch and using the pretrained model with the related works [5], [7], [8], [11]. Noisy images from DIV2K and Flickr2K created by BUIFD [18] at noise levels of 10, 15, and 20 are first used for training MRRN. After convergence, the pretrained model is used as initialized network weights for training on our dataset. Pertaining to applying transfer learning to MRRN, it has been recorded that image quality improvements have been increased by averages of 0.13 dB PSNR and 4×10^{-3} SSIM compared to training from scratch. Compared to the related works, both pretrained and training from scratch CSIE-Ms obtain 0.3–2.9 dB PSNR and 0.8×10^{-2} to 2.3×10^{-2} SSIM improvements compared to the related works.

We perform the rate-distortion (RD) curves to visualize the coding performance of CSIE-M compared to the related works
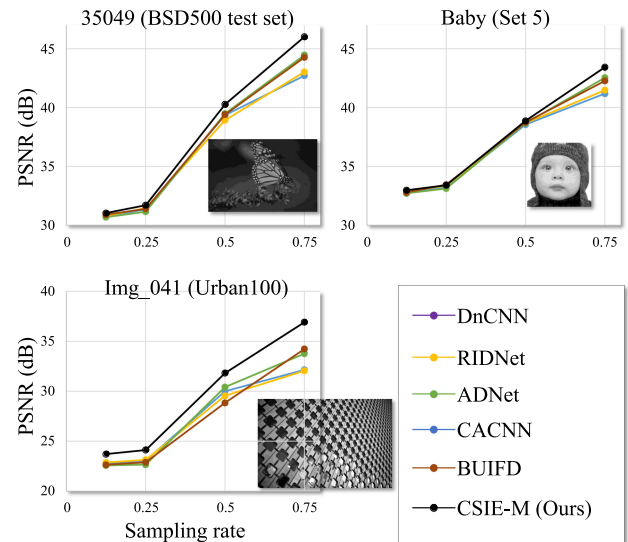

Fig. 5. RD-curves of CSIE-M and related works on sampling rates 0.125–0.75.

(Fig. 5). In these charts, the horizontal axis and vertical axis indicate the four SRs and the corresponding PSNR of the images refined by CNN. We randomly pick images from Set5, BSD500 testing set, and Urban100. It can be seen that at each SR, our proposal CSIE-M exceeds the related works in the PSNR metric. Generally, the RD curves of CSIE-M and the related approaches are distinct. In the complex scene of img_004 and img_041 of Urban100, there are significant improvements compared to the related works. Moreover, the improvement of CSIE-M is more stable than related works. In img_041, BUIFD shows third place in terms of PSNR at SRs of 0.125 and 0.25, sixth place at SR 0.5, and second place at SR 0.75.

Fig. 6 visualizes the CS enhanced images by CSIE-M and the related works under SRs of 0.125–0.75. From top to down, every two rows are images reconstructed from SRs of 0.75–0.125. It can be observed from Fig. 6 that CSIE-M returns the enhanced image with sharper edges than the related works. CSIE-M removes compression artifacts such as ringings and aliasing. At the low SRs of 0.125 and 0.25, CSIE-M sharpens the local details such as blurring edges and aliasing. At high SRs of
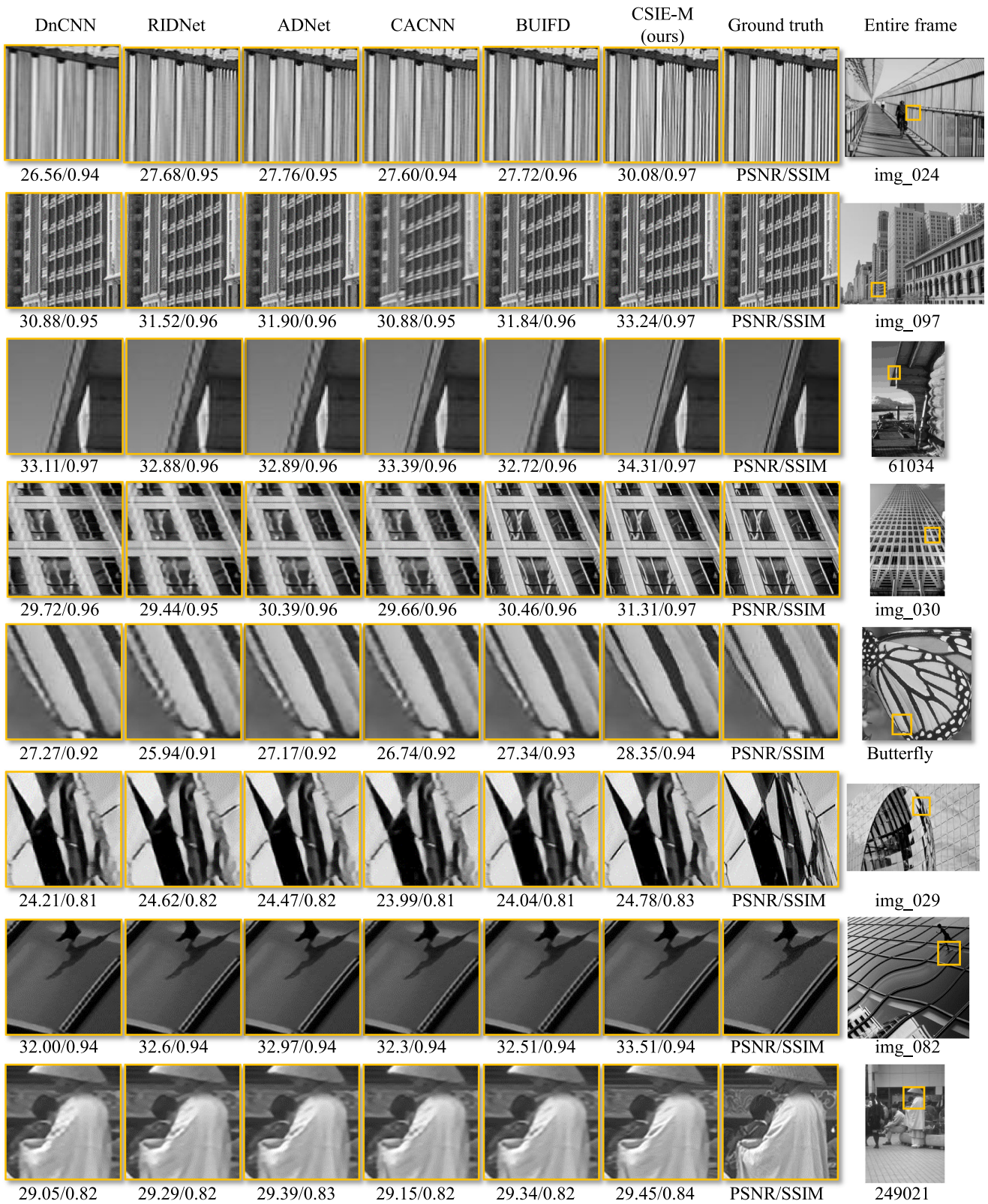
Fig. 6. Visualizations of reconstructed images enhanced by DnCNN [5], RIDNet [7], CACNN [6], ADNet [8], BUIFD [11], and CSIE-M (ours) under the sampling rates of 0.125–0.75, and the ground-truth label.

0.5 and 0.75, where more information is sent to the decoder, CSIE-M tends to produce better structures in the enhanced images.

V. Conclusion

Different from the other image compression standards, such as JPEG and JPEG2000, CS can reconstruct many images with different qualities. Using this property, this is the first time to propose a deep learning based CSIE-M. In the decoder, reconstructed images are scored and ranked by a no-reference quality ranking module before feeding to the quality enhancement module. In the quality enhancement module, low-level and high-level features extracted from CS reconstructed images were exploited and enriched by the proposed RDBlock. As a result, 1.88–8.07 dB PSNR improvements under the SRs of 0.125–0.75 were obtained. We further experimented on the effectiveness of CSIE-M with and without the no-reference quality ranking module. The result showed that 1.127 dB PSNR can be improved when using the no-reference quality enhancement module. Moreover, our framework CSIE-M, which utilizes multiple-input images to enhance the reconstructed image quality, outperformed the one-to-one learning networks. The proposal can be integrated into IoT imaging systems to enhance the CS reconstructed images, giving better visual quality for end users and a promising approach for designing AIoT systems.

References

[1] Y. Zhang, P. Wang, H. Huang, Y. Zhu, D. Xiao, and Y. Xiang, "Privacy-assured FogCS: Chaotic compressive sensing for secure industrial big image data processing in fog computing," *IEEE Trans. Ind. Informat.*, vol. 17, no. 5, pp. 3401–3411, May 2021.

[2] Y. Zhang, Q. He, G. Chen, X. Zhang, and Y. Xiang, "A low-overhead, confidentiality-assured, and authenticated data acquisition framework for IoT," *IEEE Trans. Ind. Informat.*, vol. 16, no. 12, pp. 7566–7578, Dec. 2020.

[3] Y. Zhang, P. Wang, L. Fang, X. He, H. Han, and B. Chen, "Secure transmission of compressed sampling data using edge clouds," *IEEE Trans. Ind. Informat.*, vol. 16, no. 10, pp. 6641–6651, Oct. 2020.

[4] D. L. Donoho et al., "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[5] 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, Jul. 2017.

[6] C. Jia et al., "Content-aware convolutional neural network for in-loop filtering in high efficiency video coding," *IEEE Trans. Image Process.*, vol. 28, no. 7, pp. 3343–3356, Jul. 2019.

[7] S. Anwar and N. Barnes, "Real image denoising with feature attention," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 3155–3164.

[8] C. Tian, Y. Xu, Z. Li, W. Zuo, L. Fei, and H. Liu, "Attention-guided CNN for image denoising," *Neural Netw.*, vol. 124, pp. 117–129, 2020.

[9] Z. Yue, Q. Zhao, L. Zhang, and D. Meng, "Dual adversarial network: Toward real-world noise removal and noise generation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 41–58.

[10] H. Qiu, Q. Zheng, G. Memmi, J. Lu, M. Qiu, and B. Thuraisingham, "Deep residual learning-based enhanced JPEG compression in the Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 17, no. 3, pp. 2124–2133, Mar. 2021.

[11] M. El Helou and S. Süsstrunk, "Blind universal Bayesian image denoising with Gaussian noise level learning," *IEEE Trans. Image Process.*, vol. 29, pp. 4885–4897, Mar. 2020.

[12] Z. Guan, Q. Xing, M. Xu, R. Yang, T. Liu, and Z. Wang, "MFQE 2.0: A new approach for multi-frame quality enhancement on compressed video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 3, pp. 949–963, Mar. 2021.

[13] R. Yang, M. Xu, Z. Wang, and T. Li, "Multi-frame quality enhancement for compressed video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6664–6673.

[14] E. Candes and J. Romberg, "l1-Magic: Recovery of sparse signals via convex programming," vol. 4, p. 14, 2005. [Online]. Available: www.acm.caltech.edu/l1magic/downloads/l1magic.pdf

[15] E. Van DenBerg and M. P. Friedlander, "Probing the pareto frontier for basis pursuit solutions," *SIAM J. Sci. Comput.*, vol. 31, no. 2, pp. 890–912, 2008.

[16] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

[17] T. T. Do, L. Gan, N. Nguyen, and T. D. Tran, "Sparsity adaptive matching pursuit algorithm for practical compressed sensing," in *Proc. 42nd IEEE Asilomar Conf. Signals, Syst. Comput.*, 2008, pp. 581–587.

[18] D. Vijayalakshmi, M. K. Nath, and O. P. Acharya, "A comprehensive survey on image contrast enhancement techniques in spatial domain," *Sens. Imag.*, vol. 21, no. 1, pp. 1–40, 2020.

[19] D. Połap, "An adaptive genetic algorithm as a supporting mechanism for microscopy image analysis in a cascade of convolution neural networks," *Appl. Soft Comput.*, vol. 97, 2020, Art. no. 106824.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[21] Z. Li, J. Yang, Z. Liu, X. Yang, G. Jeon, and W. Wu, "Feedback network for image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3867–3876.

[22] E. Agustsson and R. Timofte, "NTIRE 2017 challenge on single image super-resolution: Dataset and study," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2017, pp. 1122–1131.

[23] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th IEEE Int. Conf. Comput. Vis.*, 2001, pp. 416–423.

[24] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. A.Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in *Proc. Brit. Mach. Vis. Conf.*, 2012, pp. 135. 1–135. 10.

[25] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Proc. Int. Conf. Curves Surfaces*, 2010, pp. 711–730.

[26] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 5197–5206.

[27] N. Ponomarenko et al., "Image database TID2013: Peculiarities, results and perspectives," *Signal Process.: Image Commun.*, vol. 30, pp. 57–77, 2015.

[28] X. Liu, J. van de Weijer, and A. D. Bagdanov, "Rankiqa: Learning from rankings for no-reference image quality assessment," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1040–1049.

[29] J. M. Bioucas-Dias and M. A. Figueiredo, "A new twist: Two-step iterative shrinkage/thresholding algorithms for image restoration," *IEEE Trans. Image Process.*, vol. 16, no. 12, pp. 2992–3004, Dec. 2007.

[30] W. Deng, W. Yin, and Y. Zhang, "Group sparse optimization by alternating direction method," in *Wavelets Sparsity XV*, Bellingham, WA, USA: International Society for Optics and Photonics, 2013.

[31] D. Needell and J. A. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Appl. Comput. Harmon. Anal.*, vol. 26, no. 3, pp. 301–321, 2009.

[32] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, "Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 58, no. 2, pp. 1094–1121, Feb. 2012.
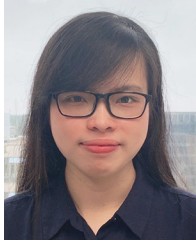
Chi Do-Kim Pham received the B.S. degree in computer science from the University of Information Technology (UIT), Vietnam National University-Ho Chi Minh City, Ho Chi Minh City, Vietnam, in 2017, and the M.E. degree in applied informatics from Hosei University, Tokyo, Japan, in 2019. She is currently working toward the Ph.D. degree in applied informatics with the Graduate School of Science and Engineering, Hosei University.

She is currently a Research Assistant with Hosei University. Specifically, she is interested in applying learning-based techniques to image and video coding standards. Her primary research interests include deep learning and image processing.

Ms. Pham is a Reviewer for *IEEE Access*.

Jian Yang received the B.S. degree in computer science and technology from the Southwest University of Science and Technology, Mianyang, China, in 2019. He is currently working toward the master's degree in applied informatics with the Graduate School of Science and Engineering, Institute of Integrated Science and Technology.

He is a Member of the Intelligent Media Processing Lab, Hosei University, Tokyo, Japan. He has been interested in machine learning and image processing since he was an undergraduate. His current research interests include image enhancement, super-resolution, and learning-based compressive sensing.
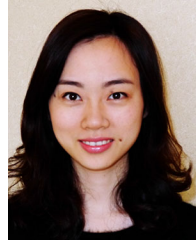
Jinjia Zhou (Member, IEEE) received the B.E. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2007, and the M.E. and D.E. degrees in system LSI from Waseda University, Tokyo, Japan, in 2010 and 2013, respectively.

She was a junior Researcher with Waseda University, Fukuoka, Japan, from 2013 to 2016. Currently, she is an Associate Professor and a Codirector of the English-based graduate program with Hosei University, Tokyo, Japan. She is also a Senior Visiting Scholar with the State Key Laboratory of ASIC & System, Fudan University, Shanghai, China. Since 2020, she has also been a specially appointed Associate Professor with Osaka University, Suita, Japan. Her research interests include algorithms and very large-scale integration architectures for multimedia signal processing.

Dr. Zhou was selected as JST PRESTO Researcher during 2017–2021. She was the recipient of the research fellowship of the Japan Society for the Promotion of Science during 2010–2013. She is a recipient of the Chinese Government Award for Outstanding Students Abroad of 2012. She was a recipient of the Hibikino Best Thesis Award in 2011. She was a corecipient of ISSCC 2016 Takuo Sugano Award for Outstanding Far-East Paper, the best student paper award of VLSI Circuits Symposium 2010, and the design contest award of ACM ISLPED 2010. She participated in the design of the world-first 8 K UHDTV video decoder chip, which was granted the 2012 Semiconductor of the Year Award of Japan. She is an Associate Editor for *IEEE Access*, an editorial board member for *Cognitive Robotics*, and a Reviewer for journals including IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I, IEEE TRANSACTIONS ON VLSI SYSTEMS, and *IEEE Transactions on Multimedia*.