# A Comprehensive Benchmark for Single Image Compression Artifact Reduction

Jiaying Liu, *Senior Member, IEEE*, Dong Liu, *Senior Member, IEEE*, Wenhan Yang, *Member, IEEE*, Sifeng Xia, *Student Member, IEEE*, Xiaoshuai Zhang, and Yuanying Dai

***Abstract*— We present a comprehensive study and evaluation of existing single image compression artifact removal algorithms using a new 4K resolution benchmark. This benchmark is called the Large-Scale Ideal Ultra high-definition 4K (LIU4K), and it includes including diversified foreground objects and background scenes with rich structures. Compression artifact removal, as a common post-processing technique, aims at alleviating undesirable artifacts, such as blockiness, ringing, and banding caused by quantization and approximation in the compression process. In this work, a systematic listing of the reviewed methods is presented based on their basic models (handcrafted models and deep networks). The main contributions and novelties of these methods are highlighted, and the main development directions are summarized, including architectures, multi-domain sources, signal structures, and new targeted units. Furthermore, based on a unified deep learning configuration (*i.e.* same training data, loss function, optimization algorithm, *etc.*), we evaluate recent deep learning-based methods based on diversified evaluation measures. The experimental results show state-of-the-art performance comparisons of existing methods based on both full-reference, non-reference, and task-driven metrics. Our survey gives a comprehensive reference source for future research on single image compression artifact removal and inspires new directions in related fields.**

***Index Terms*— Compression artifacts removal, benchmark, side information, loop filter, deep learning.**

## I. Introduction

LOSSY forms of compression, such as JPEG [1], HEVC (High Efficiency Video Coding) [2], and Advanced Video Coding (AVC) [3], have been widely used in image and video codecs to reduce information redundancy in transmission and storage processes to save bandwidth and resources. Based on human visual properties, the codecs make use of redundancies in spatial, temporal, and transform domains to provide compact approximations of encoded content. They effectively reduce the bit-rate cost but inevitably lead to unsatisfactory visual artifacts, *e.g.* blockiness, ringing, and banding. These artifacts are derived from the loss of high-frequency details during the quantization process and the discontinuities caused by block-wise batch processing. These artifacts not only degrade user visual experience, but they are also detrimental for successive image processing and computer vision tasks.

In our work, we focus on the degradation of compressed images. The degradation configurations of two codecs are considered: JPEG and HEVC. Most modern codecs first divide the whole image into blocks, which sometimes have a fixed size, *e.g.* JPEG, while others have different sizes, *e.g.* HEVC. Then, transformations, *e.g.* discrete cosine transformation (DCT) and discrete sine transformation (DST) *etc.*, follow to convert each block into transformed coefficients with more compact energy and sparser distributions than those in the spatial domain. After that, quantization is applied to the transformed coefficients, based on the pre-defined quantization steps, to remove the signal components that have less significant influence on the human visual system. The quantization intervals are usually much larger in high-frequency components than those in low-frequency components because the human visual system is less capable of distinguishing high frequency components. It is worthy of noting that the quantization step is the main cause of artifacts. After quantization, the boundaries between blocks become discontinuous. Thus, *blocking artifacts* are generated. *Blurring* is caused by the loss of high-frequency components. In regions that contain sharp edges, the *ringing* artifacts become visible. When the quantization step becomes larger, the reconstructed images suffer from severe distortions. Noticeable *banding effects* appear in smooth regions over the image.

Great efforts have been dedicated to the restoration of compressed images. Early preliminary works [4], [5] perform filtering along the boundaries to remove simple artifacts. After that, data-driven methods proposed learning the inverse mapping of compression degradations to remove artifacts. These methods serve two objectives: 1) better inference models, *e.g.*, sparse coding [6] and deep networks [7]; 2) and better priors and side information [8], [9]. In recent years, the emergence of deep learning [7] has greatly improved the restoration capacity of data-driven methods due to its excellent nonlinear modeling capacity. More advanced network architectures, *e.g.* dense residual networks [10], have been put forward and more strong side information, *e.g.* partition mask [11], has been employed for compression artifacts removal. Besides

Manuscript received September 4, 2019; revised May 16, 2020 and June 28, 2020; accepted June 28, 2020. Date of publication July 13, 2020; date of current version July 21, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant No. 2018AAA0102702, in part by the National Natural Science Foundation of China under Contract 61772043 and Contract 61772483, and in part by the Beijing Natural Science Foundation under Contract L182002. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Lisimachos P. Kondi. *(Corresponding author: Dong Liu.)*

Jiaying Liu, Wenhan Yang, Sifeng Xia, and Xiaoshuai Zhang are with the Wangxuan Institute of Computer Technology, Peking University, Beijing 100080, China (e-mail: liujiaying@pku.edu.cn; yangwenhan@pku.edu.cn; xsfatpku@pku.edu.cn; jet@pku.edu.cn).

Dong Liu and Yuanying Dai are with the CAS Key Laboratory of Technology in Geo-Spatial Information Processing and Application Systems, University of Science and Technology of China, Hefei 230027, China (e-mail: dongeliu@ustc.edu.cn; daiyy@mail.ustc.edu.cn).

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/TIP.2020.3007828

these two common factors,, there are other elements that have sizable effects on final performance, such as learning-based approaches, training configurations and protocols, training data, losses, optimization approaches, data generation, and codec details. Thus, the related changes in these factors also contribute to performance gains.

Despite promising results, there are several neglected considerations in previous methods. First, there is no unified framework to understand and sort out all previous methods. It is necessary to create a survey that compares and summarizes these methods with a simple and integrated view. Second, inconsistent experimental configurations and protocols have been employed in different works. There is a lack of benchmarking efforts for state-of-the-art algorithms in a large-scale public dataset. Finally, previous datasets do not cover 4K resolution images, which sets a barrier for comparing the performance of different methods on the recently popular ultra high-definition display devices.

Our work is directly motivated to address the above issues, and our paper makes four technical contributions:

- The first contribution of this paper is to provide a comprehensive survey of compression artifact removal methods. Our survey provides a holistic view covering most of the existing methods. Particular emphasis is placed on deep learning-based single-image compression artifact removal methods, as they offer state-of-the-art performance and exhibit flexibility for further improvements.
- We introduce a new single image compression artifact removal benchmark, called the Large-scale Ideal Ultra high-definition 4K (LIU4K) dataset. It is the first dataset that includes 4K images as training and testing images for image restoration. It is also more large-scale than other existing datasets that include high-definition images. Our LIU4K provides a better foundation to evaluate performance of different methods, especially on recent popular ultra high-resolution display devices.
- We conduct a systematic and extensive range of experiments to compare state-of-the-art methods quantitatively with diversified measures. In our experiments, we contrast the new LIU4K dataset as well as previous commonly used datasets with a unified experimental setting, including the same training data, optimization method, and loss function *et al*. Thorough evaluations and analyses show the performance and limitations of state-of-the-art methods. New rich insights inspire novel research directions.
- We also explore generalizing some constraints and training strategies from JPEG artifact removal to general compression artifact removal. Three strategies, including dense DCT transform constraints, mixed batches with different patch sizes, and gradually expanding patch sizes are used in our experimental setting. These strategies also benefit future compression artifacts removal methods.

## II. A New Dataset for Restoration: LIU4K

### A. Previous Datasets

We first review existing testing and training datasets: 1) Testing: *BSD100*, *Kodak*, *DIV2K-test*, *Set5*, *Set14*, *Classic5*, and *Twitter*; 2) Training: *BSD400*, *DIV2K-train*, and *Mini-ImageNet*.

TABLE I

The Summary of Different Datasets for Compression Artifact Removal

| Dataset | Number | Resolution | Train/Test | Features |
|---|---|---|---|---|
| Kodak | 24 | 768×512 | Test | Earliest Milestone |
| Set5 | 5 | 500 × 500 | Test | Small and Effective |
| Set14 | 14 | 250 × 250 - 500 × 500 | Test | Small, Effective |
| Classic5 | 5 | 512 × 512 | Test | Small, Effective |
| BSD500 | 200/200/100 | 321 × 481 | Train / Test Validation | Middle Scale with Abundant Texture |
| Mini-ImageNet | 300,000 | 50×50-4,000×3,000 | Train | Very Large Scale |
| Twitter | 40 | 600 × 450 | Test | Complex Degradation |
| DIV2K | 800/100/100 | 2,040 × 1,000 | Train & Test Validation | Large-Scale with 2K Images |
| LIU4K | 1,500/200/80 | 3,264 × 2,448 | Train & Test Validation | Large-Scale with 4K Images |

*1) Kodak*[1]*:* This is a very representative dataset proposed in 1991, which includes 24 digital color images extracted from a wide range of films. After Kodak's creation, many image processing methods have been proposed, optimized, and evaluated based on this dataset. The image resolution is $768 \times 512$ or $512 \times 768$.

*2) BSD400 and BSD100:* These two datasets are two parts of BSD500 [12], which was originally designed for semantic segmentation. These datasets cover a wide variety of real-life scenes, with 200 training images, 200 validation images, and 100 testing images. The image resolution is $321 \times 481$ or $481 \times 321$. For image restoration, we combine the training and validation sets from BSD500 as the training set for restoration and use its testing set for the restoration evaluation.

*3) DIV2K [13]:* This dataset contains 1,000 images with a resolution of 2K. It includes 800 images for training, 100 images for validation, and 100 images for testing. The sizes of the images are around $2000 \times 1000$ or $1000 \times 2000$. The max length of the height and width of an image is 2,040, and the other one is greater than 1,000. DIV2K is a milestone dataset for image super-resolution, and it supports the NTIRE Challenge,[2] which uncovers preludes of challenges in low-level image enhancement.

*4) Set5 [14] and Set14 [15]:* These are two effective small-scale datasets for evaluating image restoration quality, and they usually provide consistent evaluation results similar to large-scale datasets. The resolution of *Set5* is less than $500 \times 500$. The image size of *Set14* is greater than $250 \times 250$ and less than $500 \times 500$.

*5) Classic5 [16]:* The *Classic5* dataset includes five represented images used for evaluating compressed image restoration. Image resolutions are $512 \times 512$.

*6) Twitter [7]:* This dataset contains 40 images compressed by the Twitter platform with sizes that vary from $3{,}264 \times 2{,}448$ to $600 \times 450$. The included artifacts are highly complex because the compression process includes a rescaling operation.

*7) Mini-ImageNet [17]:* This dataset was used to train SRGAN in [17], and it includes 300,000 images sampled from ImageNet. The small-est size is less than $50 \times 50$, and the maximum size is larger than $4{,}000 \times 3.000$. Although this dataset might lead to superior performance of restoration models, it is very time and resource-consuming to train with it.

A summary of all these datasets is provided in Table I.

[1] http://r0k.us/graphics/kodak/

[2] http://www.vision.ee.ethz.ch/ntire17/

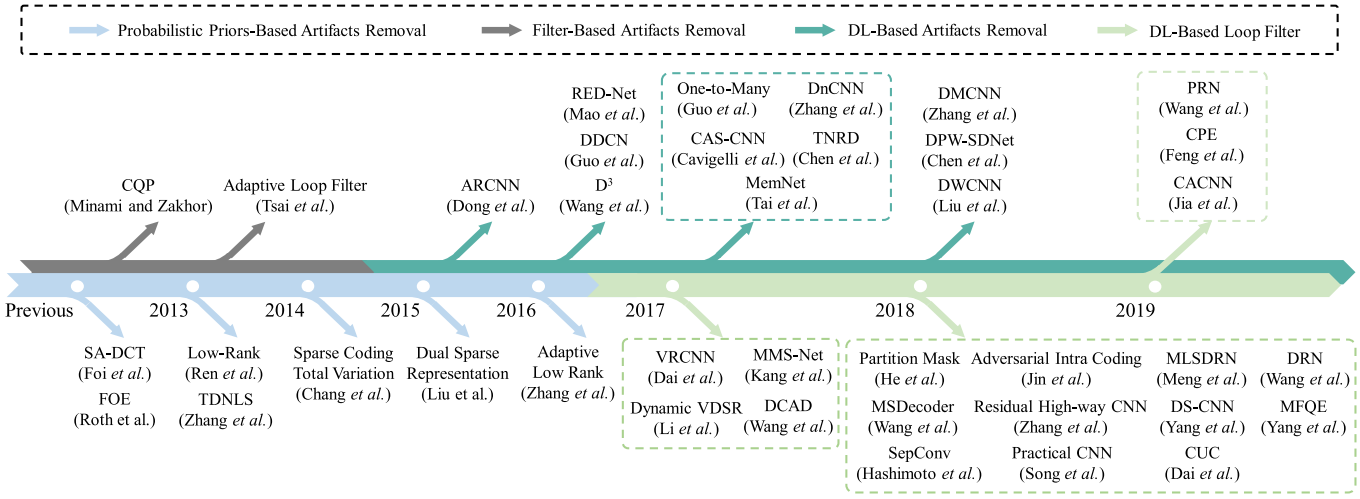
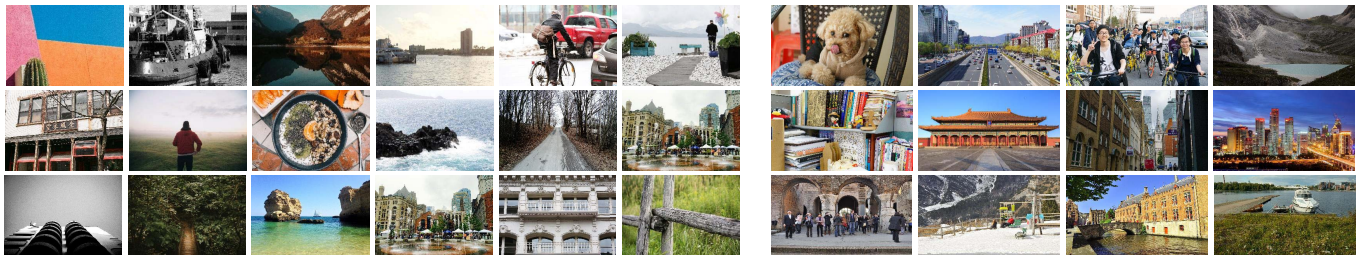



Fig. 1. Milestones in the history of compressed image restoration methods, including filter based artifact removal, probabilistic priors-based artifact removal, deep learning-based artifact removal, and deep learning-based loop filters. The time period up to 2015 was dominated by handcrafted methods, including filter-based and probabilistic priors-based artifact removal. The emergence of ARCNN [7] changed the development of this domain. A turning point is observed in 2015. After that, deep learning-based methods played a major role in the next several years. The years 2017 and 2018 welcomed a blossoming in the development of deep learning-based artifact removal and loop filters.



(a) (b)

Fig. 2. Example images sampled from LIU4K. (a) Training set. (b) Testing set.

### *B. LIU4K Dataset*

The main characteristics of the LIU4K dataset and previous datasets for image restoration are listed in Table II. LIU4K has several unprecedented superiorities as follows,

- **High-resolution definition**. Compared to previous datasets, the resolution of the images in our dataset is 2848 × 4288, which is larger than those in previous datasets, thereby offering abundant materials for testing and evaluating the performance on 4K/8K display devices.
- **Large-scale**. Our dataset is large-scale. Our training, testing, and validation images include 1,500, 200, and 80 4K images, which is much more than in previous datasets. Thus, training and evaluation processes based on LIU4K are more comprehensive and balanced.
- **Diversified and complex signals**. As shown in Table II, our dataset achieves the best results in terms of entropy-driven non-reference metrics, which demonstrates its signal diversity and complexity.
- **High visual quality**. LIU4K wins in general purpose non-reference metrics (except for Kodak and LIVE1, the training sets for some metrics), as shown in Table II, thereby confirming its high visual quality.

Training and validation data is downloaded from Pexels website.[3] The testing images come from two sources; (1) 25 images in the testing data are captured by ourselves. The cameras used to capture the 25 images include Canon EOS 5D Mark IV, Sony ILCE-6000, Canon EOS 6D, and NIKON D810. The lenses include EF 16-35mm f/4, EF 70-200mm f/2.8, EF 50mm f/1.8, Sigma 30mm F1.4 DC DN, Sony E 55-210 F4.5-6.3, EF 16-35mm f/4, Nikon 18-36mm f/3.5-4.5, and Nikon 35mm f/4. (2) The other 175 images in the testing data come from the RAISE dataset.[4] These 175 images are captured by Nikon D90, Nikon D7000, and Nikon D40 with lens VR 18-105mm f/3.5-5.6G, 35mm f/1.8G, 18-55mm f/3.5-5.6G, and 35mm f/1.8G. All images are shot in RAW format and processed by Adobe Photoshop Lightroom. The exported images are stored in lossless PNG format, and cropped to 3840 × 2160.

We perform statistical comparisons to demonstrate the superiority of the LIU4K dataset. Entropy, BPP (Bits Per Pixel), and PPI (Pixels Per Image) are used to indicate the amount of information included in each dataset. Three non-reference image quality assessment metrics are utilized to assess the perceptual image quality, including Entropy, Natural Image Quality Evaluator (NIQE) [55], Blind/Referenceless Image Spatial Quality Evaluator (BRISQE) [56], and ENtropy-based Image Quality Assessment (ENIQA) [57]. Entropy is estimated following the most primitive calculation based on per-pixel independent distribution [70]. The bits used to calculate BPP values are estimated by compressing the gray

[3]https://www.pexels.com/

[4]http://loki.disi.unitn.it/RAISE/

TABLE II

THE STATISTICAL COMPARISONS OF DIFFERENT TESTING SETS. NUMBERS IN BRACKETS DENOTE VARIANCE. SUPERIOR RESULTS ARE DENOTED IN BOLD

| Dataset | BPP | PPI ($10^5$) | Number | ENTROPY | BRISQUE | ENIQA ($10^{-4}$) | NIQE |
|---|---|---|---|---|---|---|---|
| Set5 | 0.52 (0.004) | 1.13 | 5 | 6.84 (0.548) | 33.58 (47.82) | 0.1393 (74.75) | 4.79 (4.62) |
| Classic5 | 0.65 (0.004) | 2.62 | 5 | 7.37 (0.037) | 22.99 (146.58) | 0.3696 (3.46) | 5.08 (1.62) |
| Kodak | 0.56 (0.006) | 3.93 | 24 | 6.93 (0.175) | 6.22 (17.32) | 0.0202 (6.35) | 2.95 (0.21) |
| LIVE1 | 0.58 (0.007) | 3.57 | 39 | 7.14 (0.107) | **5.01** (11.82) | 0.0218 (5.80) | 2.87 (0.18) |
| Set14 | 0.58 (0.013) | 2.30 | 14 | 6.74 (0.605) | 26.29 (147.77) | 0.1421 (194.00) | 4.71 (2.69) |
| BSD100 | 0.60 (0.010) | 1.54 | 100 | 6.94 (0.258) | 20.01 (84.39) | 0.0801 (97.13) | 3.09 (0.76) |
| DIV2K | 0.53 (0.011) | 28.35 | 100 | 7.02 (0.748) | 23.64 (131.93) | 0.0925 (47.01) | 3.18 (2.05) |
| LIU4K | **0.92** (0.0298) | **132.79** | **200** | **7.43** (0.039) | 15.98 (73.86) | **0.0036** (32.02) | **2.39** (0.24) |

TABLE III

AN OVERVIEW OF EXISTING WORKS ON NON-DEEP JPEG ARTIFACTS REMOVAL

| Method | Published | Category | Inference Model | Priors / Side Information | Basic Idea |
|---|---|---|---|---|---|
| Minami and Zakhor | TCSVT-1995 [18] | Filter | Linear model (constrained quadratic programming) | Mean squared difference of slope (MSDS) | Reduce the expected MSDS |
| Deblocking Fliter | TCSVT-2012 [69] | Filter | Hand-crafted metrics based boundary classification | Coding information (PU/TU, intra mode, motion vector *etc.*) | Divide blocking boundaries into different types and accordingly choose different kinds of deblocking filters. |
| Field of Expert | TIP-2007 [20] | Probabilistic prior | MAP framework High-order Markov model | Quantization tables | Original images are modeled as high order MRFs with learned potential functions |
| Transform Domain Non-Local Similarity | ICME-2012 [21] TIP-2013 [22] | Probabilistic prior | MAP framework Adaptive parameter selection | Nonlocal similarity | Decoded coefficients and their nonlocal estimations are fused adaptively |
| Low-Rank Minimization | DCC-2013 [23] | Probabilistic prior | Patch clustering Singular value thresholding | Local sparsity Low-rank prior | Similar patches are clustered and reconstructed by low-rank minimization |
| Sparse Coding with Total Variation | TSP-2014 [6] | Probabilistic prior | Sparse coding | Sparsity prior Total variations | Combination of sparse representation and total variations |
| Dual Domain Sparse Representation | CVPR-2015 [24] TIP-2016 [25] | Probabilistic prior | Sparse coding | Spatial domain DCT domain External data | Sparse representations jointly in dual domains augmented by external data |
| SA-DCT Transform | TIP-2007 [16] | Probabilistic prior | Wiener filtering in SA-DCT domain | SA-DCT transform Structural constraint | Transforms use adaptive supports which leads to better edge reconstruction |
| Adaptive Low-Rank Minimization | TIP-2016 [26] | Probabilistic prior | Patch clustering Singular value thresholding | Local sparsity Low-rank prior Transform coefficient variance Quantization step | Thresholds in SVT are adaptively determined |

version of an image into a PNG image. The work in [58] has shown that, the non-reference image quality assessment metrics are highly correlated to human perception and are superior to some full-referenced measures in terms of visual quality. In our work, we calculate values for NIQE, BRISQE, and ENIQA with the codes provided by their authors using the default settings. For NIQE, BRISQE, and ENIQA, small values indicate better image quality.

As seen in Table II, LIU4K has a larger scale than previous datasets. From the perspective of information theory, the images in LIU4K are more informative; its mean BPP and entropy values are greater, which means that the dataset contains more information. For perceptual image quality assessment, LIU4K also achieves very competitive scores in BRISQUE, ENIQA, and NIQE. Note that the values of BRISQUE and ENIQA in LIU4K are worse than those of Kodak and LIVE1, since BRISQUE and ENIQA are trained on the TID [76] and LIVE1 [75] datasets, respectively. The three datasets, TID, LIVE1, and Kodak, share many of the same images, which naturally leads to the undistorted images in Kodak and LIVE1 having very good scores that benefit the overall dataset scores. In general, these assessments indicate that images in LIU4K are of relatively high perceptual quality and suitable for image restoration tasks.

## III. ALGORITHM SURVEY

Approaches designed for compression artifact removal, namely loop filters in codecs, have been proposed in the body of literature. There are four categories in our review: filter-based methods, probabilistic prior-based methods, deep learning-based JPEG artifacts removal methods, and deep learning-based loop filter methods. The first and last two categories are summarized in Table III and IV, respectively. We review the four categories and then briefly summarize their technical improvements. Note that, the technologies discussed in our work can be applied without changing the existing codec pipeline.

JPEG is now the most widely used standard for natural image compression, but its compression efficiency is not state-of-the-art. HEVC and its variants (BPG and HEIF) represent the highest standards of natural image/video compression. Both coding standards employ block-wise compression schemes, which are the primary causes of blockings. Most previous works are based on these two standards and their related implementations, such as focusing on JPEG artifact reduction [7], [8], [27], [28] (the upper half of Table IV) and loop filters [36]–[40] (the bottom part of Table IV). Therefore, in our benchmark paper, we hope to summarize previous developments and compare different methods in a comprehensive and fair manner. Therefore, JPEG and HEVC are considered in our paper.

### A. Filtering-Based Methods

The earliest methods [46], [47] perform filtering operations to remove compression artifacts. Later approaches [2], [18]

TABLE IV
AN OVERVIEW OF EXISTING WORKS ON DEEP COMPRESSION ARTIFACT REMOVAL

| Method | Published | Inference Model | Priors / Side Information | Basic Idea |
|---|---|---|---|---|
| Artifact Reduction CNN | ICCV-2015 [7] | Three-layer CNN | / | The first work introducing deep models to the topic |
| Trainable Nonlinear Reaction Diffusion | TPAMI-2017 [27] | Trainable nonlinear diffusion model | / | The proposed nonlinear diffusion model unrolling into a deep network |
| $D^3$ Model | CVPR-2016 [8] | Learned iterative shrinkage and thresholding with DCT layers | DCT domain constraint Sparsity constraint | The proposed nonlinear diffusion model unrolling into a deep network |
| Denoising CNN | TIP-2017 [28] | CNN with residual learning and batch normalization | / | The combination of residual learning, batch normalization, and Adam optimization |
| Dual-Domain CNN | ECCV-2016 [9] | A two-branch CNN | Range of DCT coefficients | A two-branch CNN works in pixel and DCT domains and finally aggregates their information |
| Residual Encoder-Decoder Network | NIPS-2016 [29] | Encoder-decoder with skip connections | / | An encoder-decoder with symmetric skip connections |
| Compression Artifact Suppression CNN | IJCNN-2017 [30] | Encoder-decoder with skip connections | Multi-scale losses | An encoder-decoder constrained by multi-scale losses |
| One-to-Many Network | CVPR-2017 [31] | ResNet Shift-and-average strategy | Perceptual loss Adversarial loss JPEG loss | A ResNet takes input as random noise and a compressed image, and its output is constrained by three losses |
| MemNet | ICCV-2017 [32] | DenseNet architecture Memory block | Multi-supervision Long-term memory | The network is stacked by memory blocks, consisting of a recursive unit and a gate unit, to learn explicit persistent memories |
| DMCNN | ICIP-2018 [33] | A two-branch auto-encoder with dilated convolution | DCT domain constraint Multi-scale loss | It integrates the dual domain architecture (DCT and spatial domains), DCT loss and multi-scale loss |
| Multi-level Wavelet-CNN | CVPRW-2018 [34] | Encoder-decoder with Wavelet transforms | Wavelet signal structure | Wavelet transforms are introduced into CNN architecture |
| Dual Pixel-Wavelet Domain Deep CNN | CVPRW-2018 [35] | A two-branch CNN with Wavelet transforms | Dual domains Wavelet signal structure | A two-branch CNN is constructed to make use of both redudenancy in pixel and frequency domains |
| VRCNN | MMM-2017 [36] | Variable-filter-size Residue-learning | / | The designed CNN owns variable filter size to learn the residual between input and target frames |
| Deep CNN-based Auto Decoder | DCC-2017 [37] | ResNet | TU size | A ResNet is used for quality enhancement in the decoder end |
| Partition Mask CNN | ICIP-2018 [11] | ResNet | CU size | The CU size is utilized and integrated with distorted decoded frame |
| Residual High-way CNN | TIP-2018 [38] | Highway network | QP range | Residual highway CNNs trained delicately for each QP range |
| MLSDRN | DCC-2018 [39] | Multi-channel long-short term dependency residual network | Block boundary Multi-channel | MLSDRN uses an update cell to adaptively store and select the long-term and short-term dependency |
| Adversarial Intra Coding | ICASSP-2018 [40] | Multi-scale structure | Adversarial learning | A multi-level progressive refinement network with adversarial learning |
| Decoder-Side Scalable CNN | ICME-2017 [41] | Two-branch scalable CNN | / | The network has two branches. A group of switches controls whether the complicated one is activated. |
| Practical CNN | ICIP-2018 [42] | Compressed fixed point CNN | QP | The network also takes QP as input. After training, the model is compressed and concerted into fixed point format. |
| Multi-Scale Deep Decoder | DCC-2018 [43] | CNN Multi-scale LSTM | / | Each frame is fed into a CNN, then a multi-scale LSTM is connected to fuse multi-frame redundancies. |
| MF Quality Enhancer | CVPR-2018 [44] | SVM classier CNN-based alignment CNN-based enhancer | Neighboring peak quality frames | The neighboring high-quality frames are fed into a CNN to facilitate inference of enhanced frames. |
| Separable CNN filter | JVET-K0158 [45] | SE block Separable convolution | Normalized Y/U/V Normalized QP | The network takes as input Normalized Y/U/V and QP and consists of SE blocks and separable convolutions. |
| Dense Residual Network | VCIP-2018 [10] | Dense shortcuts Residual learning Bottleneck layer | / | The network consists of dense shortcuts, residual learning, and bottleneck layers. |
| CU Classification | VCIP-2018 [72] | Multiple variable-filter-size residue-learning | CU classification | A classifier is employed to decide whether to use VRCNN-ext for each coding unit. |
| Progressive Rethinking Network | ICIP-2019 [54] | Progressive Rethinking Block and Network | Multi-scale mean value of CUs | The progressive rethinking network is built to take multi-scale mean value of coding units as side information. |
| Coding Prior-based High Efficiency Restoration | ICIP-2019 [71] | Weight Normalization | Unfiltered frame Prediction frame | An EDSR-like network takes the unfiltered and prediction frames as side information and is trained with weight normalization. |
| Content-Aware CNN | TIP-2019 [73] | Context-based model selection | Clusters based on quality ranking | The discriminative model is learned to analyze the region content for model selection. An iterative training is proposed to label filter categories and fine-tune CNN models. |

attempt to infer the parameters of filtering operations adaptively. Minami and Zakhor [18] observed that quantizing the DCT coefficients of two neighboring blocks increases the expected value of the mean squared difference of slope (MSDS) between the slopes across two adjacent blocks, and the average value of the boundary slopes from each of the two blocks. Thus, a constrained quadratic programming problem is built to reduce the expected value of this MSDS to decrease the blocking effects while preserving texture details. In HEVC, an in-loop deblocking filter is specially designed [69] to reduce the blocking artifacts between coding units. The picture is divided into $8 \times 8$ blocks, and boundaries on the $8 \times 8$ grid are classified by a series of metrics. Different levels of deblocking operations are later performed on the boundaries according to their types.

### *B. Probabilistic-Prior Methods*

Some successive approaches are based on probability estimations of image-prior models. Based on their basic models, these methods can be further categorized into Markov random fields [20], non-local similarities [21], [22], low-rank

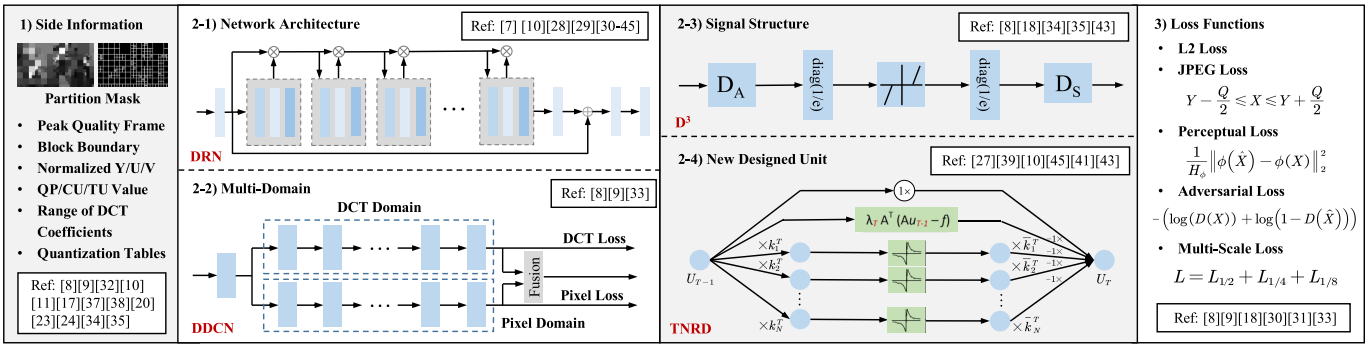



Fig. 3. The technical improvement pathway for deep learning-based compression artifacts removal and codec loop filters.

minimizations [22], [23], sparse codings [6], [24], [25], and adaptive DCT transformations [16]. In [20], the distortion term is modeled as additive, spatially correlated Gaussian noise, and the original image is depicted as a high order Markov random field based on the fields-of-experts framework. Non-local based methods [21], [22] consider similar blocks to be potentially correlated, estimate the overlapped-block transform coefficients, and remove compression noise from non-local similar blocks. For low-rank based methods, Ren *et al.* [23] performed patch clustering and low-rank minimization simultaneously to make use of both local sparsity and non-local similarity. A later work [22] selects thresholds adaptively for each group of similar patches based on compression noise levels and decomposed singular values. In [16], a new shape adaptive DCT transform is proposed for image compression artifact reduction.

### C. Deep Learning-Based JPEG Artifacts Removal

Deep learning-based methods largely improve the restoration capacity of data-driven methods. ARCNN [7] is a seminal work and adopts the architecture of a three-layer CNN. Deep Dual-Domain ($D^3$) [8] is the first work to introduce the DCT-domain priors to facilitate JPEG artifacts removal. It combines both the strong learning capacity of deep networks, as well as the problem-specific knowledge of JPEG artifact removal.

Successive works fall into two main streams: better network architectures [28], [30], [32] and better utilization of DCT domain information [9], [33]. Many advanced networks are constructed to model the rich dependencies of deep features. The Residual Encoder-Decoder Network (RED-Net) [29] and Compression Artifact Suppression CNN (CAS-CNN) [30] utilize deep encoding-decoding frameworks with symmetric convolutional-deconvolutional layers. Tai *et al.* [32] constructed a deep persistent memory network. Memory blocks consist of a recursive unit and a gate unit to retain memories. The former extracts multi-level representations from the last input feature while the latter learns to control the ratio between the memory and current input. Dual-Domain Multi-Scale CNN (DMCNN) [33] integrates the dual domain and auto-encoder style networks with dilated convolutions to create extensive receptive fields and eliminate banding effects. In [34], wavelet transforms were introduced into CNN architectures for a better trade-off between receptive field size and computational efficiency. In [35], a two-branch CNN handles the restoration in the pixel and discrete wavelet domains.

Besides network improvement, some works try to embed traditional priors or constraints into deep networks, *e.g.* sparsity [8], nonlinear diffusion [27], multi-scale constraints [30], [33], and wavelet signal structures [34], [35]. In one-to-many networks [31], adversarial learning is introduced to facilitate visually pleasing restoration results. A performance comparison of typical deep learning-based JPEG artifact removal processes featured in the aforementioned research works is presented in Fig. 5.

### D. Deep Learning-Based Loop Filters

Besides JPEG, deep learning techniques have also been applied to the latest codecs, *e.g.* HVEC, as a post-processor. Beyond the improvements embodied in JPEG artifact removal, deep-learning based loop filters focus more on handling the degradation caused by variable-size partitions and utilizing side information from codecs. Variable-Filter-Size Residue-Learning CNN (VRCNN) [36] is a pioneering work. The designed CNN owns variable filter sizes to learn the residual between input and target frames. Successive works also fall into two classes: those with better networks and those with better side information. Zhang *et al.* [38] proposed a residual highway convolutional neural network (RHCNN) for in-loop filters of HEVC. In [43], Wang *et al.* proposed a multi-scale LSTM to fuse multi-frame redundancies along a temporal dimension to acquire fused features. Meng *et al.* [39] proposed a multi-channel long-short term dependency residual network to simulate the mechanism of human memory updating and introduced an update cell, which learns to store and select long-term and short-term dependencies adaptively. Li *et al.* [52] presented a dynamic classification mechanism. An up-to-one byte flag indicates the complexity of video content and the quality of each frame. In [41], Yang *et al.* designed a scalable deep CNN to reduce distortion of both I and B/P frames in HEVC. It has two branches and a group of switches to control whether a DS-CNN-B branch is activated based on the resource state. In [42], Song *et al.* developed a CNN that can enhance compressed videos of different qualities with low redundancy. In [44], Yang *et al.* enhanced compressed video frames using neighboring high-quality frames. A novel multi-frame convolutional neural network is built for compressed video enhancement. In [45], Hashimoto *et al.* proposed a CNN with squeeze and excitation blocks and spatial separable convolution for deblocking. In [10], Wang *et al.* proposed a dense residual convolutional neural network (DRN). In this network, dense shortcuts and residual learning are combined. Bottleneck layers are injected into each DRN to save

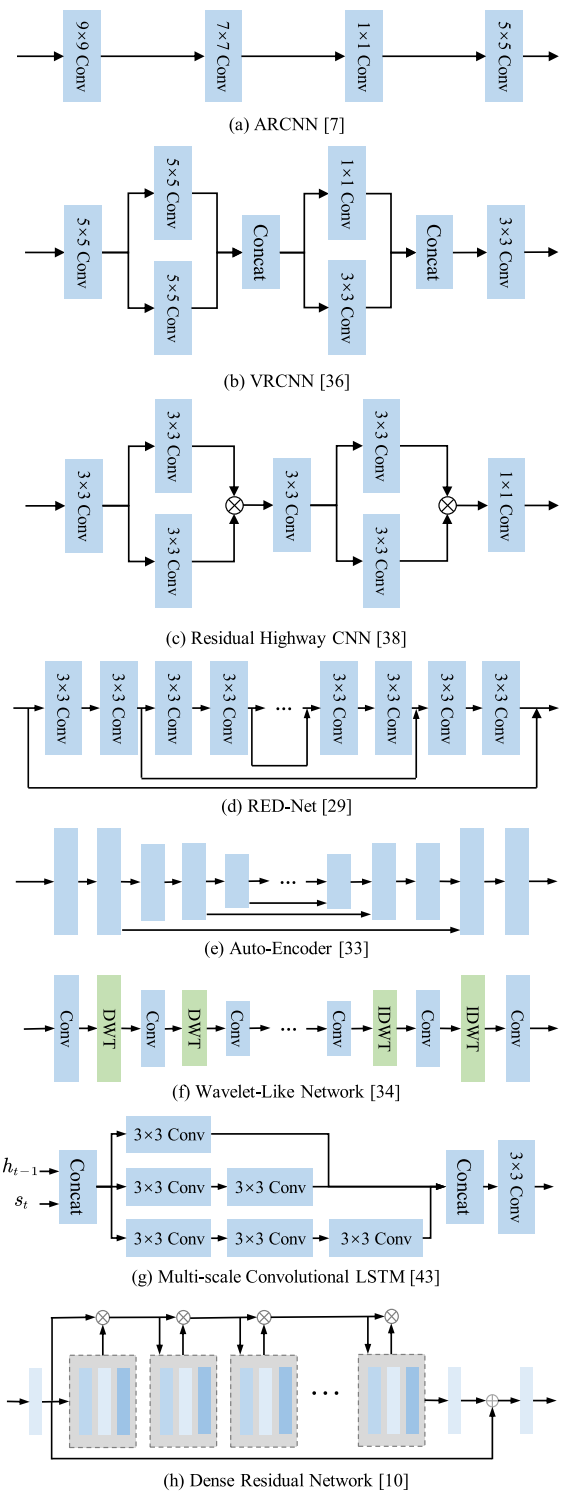



Fig. 4. The network improvement routes for compression artifacts removal and loop filters of codecs, where the multiplication sign in the circle in (c) denotes the element-wise multiplication operation.

computational resources while adaptively fusing hierarchical features.

Various kinds of side information have been designed for more effective post-processing of compression artifact reduction. This side information includes: compression parameters from coding tree units (CTU) [51], partition masks of CTU [11], QP parameters [38], block boundaries [39],

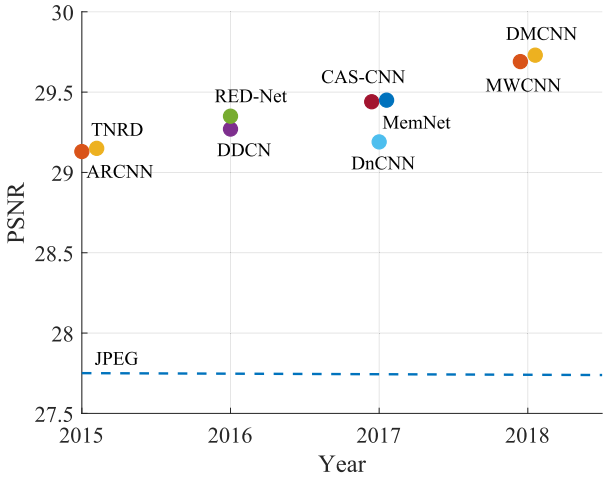



Fig. 5. Recent evolution of deep learning-based JPEG artifacts removal. We can observe significant performance (PSNR) improvements since deep learning entered the scene in 2015. The performances shown here are directly quoted from the published papers.

complexities [52], peak quality frames and optical flow [44], and normalized Y/U/V and normalized QP [45], *etc*.

### *E. Technical Improvement Summary*

The typical improvement pathway for deep learning-based compression artifact reduction is summarized in Fig. 3. Three aspects of improvements are included: *side information utilization*, *e.g.* injecting a partition mask from CTU [11] as input; *network improvement*, *e.g.* a dense residual network [10]; and *novel loss function*, *e.g.* adversarial loss [31]. For network improvement, all methods are improved in four facets: 1) *network architecture improvement* (summarized more specifically in Fig. 4); 2) *multi-domain networks*, *e.g.* DMCNN [33]; 3) *signal structure embedding*, *e.g.* D3 [8]; 4) *new unit designs*, *e.g.* TNRD [27]. In the next section, we benchmark these methods using unified protocols.

## IV. Algorithm Benchmarking

With the rich resources provided by LIU4K, we evaluate nine representative state-of-the-art algorithms: Shape-Adaptive DCT (SA-DCT) [16], Artifacts Removal CNN (ARCNN) [7], Trainable Nonlinear Reaction Diffusion (TNRD) [27], Denoising CNN (DnCNN) [28], Persistent Memory Network (MemNet) [32], Dual-Domain Convolution Network (DDCN) [9], One-To-Many Network (OTM) [31], Dual-domain Multi-scale CNN (DMCNN) [33], Multi-Level Wavelet-CNN (MWCNN) [34], Variable-filter-size Residue-learning CNN (VRCNN) [36], and Progressive Rethinking Network (PRN) [3]. Our selected baselines cover most of the representative methods. SA-DCT is a traditional non-deep method. The successive six methods are deep learning-based JPEG artifact reduction methods. The last two are deep learning-based loop filter methods. We apply most learning-based methods to restore the images compressed by JPEG and HEVC. For JPEG artifact reduction, we train the models on the training of LIU4K. For loop filters, the models are trained on the training sets of both BSD500 and LIU4K. During our training phase, we use 80 additional 4K images as our LIU4K validation set. Note that, the source codes of

TABLE V
SPECIFIC SETTINGS OF OUR IMPLEMENTED ARCNN

| Module | Settings |
|---|---|
| 1st Conv | Output Ch: 64, Kernel Size: 9, Pad: 4 |
| 2nd Conv | Output Ch: 32, Kernel Size: 7, Pad: 3 |
| 3rd Conv | Output Ch: 16, Kernel Size: 5, Pad: 2 |
| 4th Conv | Output Ch: 1, Kernel Size: 3, Pad: 1 |
| Output | Residue |
| Activation | PReLU [75] |

SA-DCT and TNRD provided by the authors only support removing JPEG artifacts with quality factors of 10, 20, 30, 40 and 10, 20, 30, respectively. Thus, for these two methods, we only compare their performances in these cases.

We also add residual learning in our implemented ARCNN for fast training and comparison. The network consists of four convolutional layers. In the first convolutional layer, the channel number of the output feature is 64, and the convolutional kernel size is 9 with a padding number of 4. In the second convolutional layer, the channel number of the output feature is 32, the convolutional kernel size is 7, and the padding number is set to 3. In the third convolutional layer, the channel number of the output feature is 16, and the convolutional kernel size is 1. The last convolution's output channel number is 1, and the kernel size is 5 with a padding number of 2. PReLU [74] is used as the activation function. The network aims to predict the residue, which is the difference between the compressed and original images. The settings and configurations are briefly summarized in Table V. For other configurations, we follow ARCNN's original settings [7].

### A. Advanced Training Strategies

In our benchmarking, we also make efforts to extend some constraints and methods of JPEG artifact reduction to the general compression artifacts reduction.

*1) Variable Block-Size DCT Domain Constraints:* The JPEG codec always partitions an image into $8 \times 8$ blocks and then performs transformation and quantization block by block. For some codecs, *e.g.* HEVC, the partitioned block sizes are not the same. Thus, the original DCT branch constraint that regularizes reconstruction of fixed-sized $8 \times 8$ blocks in JPEG artifacts removal might not be reasonable. With this in mind, we change the DCT constraint design to adapt to variable block-size partition structures used in HEVC codecs. We extend the DCT branch into two branches, as shown in Fig. 6. Given a compressed image $I_c$, one of the DCT branches transforms $I_c$ with the $8 \times 8$ DCT transform, refines the transformed signal in the DCT domain with the auto-encoder and then projects the signals back to the image domain via an inverse $8 \times 8$ DCT layer (iDCT layer) to obtain $\widetilde{I}^{8}_{\mathrm{DCT}}$. The other DCT branch does the same thing but with the $16 \times 16$ DCT and iDCT layers to obtain $\widetilde{I}^{16}_{\mathrm{DCT}}$. Therefore, each DCT branch is responsible for constructing DCT domain constraints at certain spatial patch sizes. After that, the compressed image and two outputs of the DCT branches are concatenated together $\left[I_c, \widetilde{I}^{8}_{\mathrm{DCT}}, \widetilde{I}^{16}_{\mathrm{DCT}}\right]$ as the input of the pixel-domain auto-encoder to generate the residual image $I_r$. Finally, the restored image is obtained via: $I_s = I_r + I_c$. In this way, the restoration process makes full use of the signal characteristics of different spatial patch sizes in the DCT domains to better infer restored images.

*2) Gradually Expanding Patch Sizes:* DCT branches are not stable during training. To make our training more effective, we first utilize small patches to train our network and then enlarge the patch size gradually. We use $P_{\mathrm{JPEG}}$ and $P_{\mathrm{HEVC}}$ to denote the patch sizes for training artifact reduction models for JPEG and HEVC, respectively. $e$ denotes the epoch number. The sizes of the training patches used to train models to alleviate JPEG artifacts are set as follows,

$$P_{\mathrm{JPEG}} = \begin{cases} 56, & e \in [1, 6], \\ 112, & e \in [7, 9], \\ 168, & e \in [10, 12], \\ 224, & e \in [13, 15], \\ 256, & e \in [16, 60]. \end{cases} \tag{1}$$

For HEVC post-processing, all interval bounds for $e$ are multiplied by 5. Therefore, we have:

$$P_{\mathrm{HEVC}} = \begin{cases} 56, & e \in [1, 34], \\ 112, & e \in [35, 49], \\ 168, & e \in [50, 64], \\ 224, & e \in [65, 79], \\ 256, & e \in [80, 300]. \end{cases} \tag{2}$$

This strategy leads to a better constraint in the DCT branch and also leads to better performance.

*3) Learning With Mixed Batches:* For methods with high complexities, it is impossible to train a model with both large patch sizes and large batch sizes at the same time. However, both sides are important for training a good artifact reduction model. A large patch enables a model to make use of information from a large context. A large batch size is capable of providing a diverse context and reasonable gradient descend directions during the training phase. To achieve both goals, we propose applying training with mixed-batches, *i.e.* combination of large patch, small batch and small patch, large batch.

In our implementation, given a batch size of 30, one sub-batch's batch size is set to 2, with a patch size based on Eqn. (1) and (2). The other's batch size is set to 28, with a patch size set to 32 constantly. In this way, with limited GPU memory resources, network training is stabilized by using a large batch size, and at the same time, the model also learns information from a large context with large patch size. In our benchmark, we train MemNet and PRN in this way.

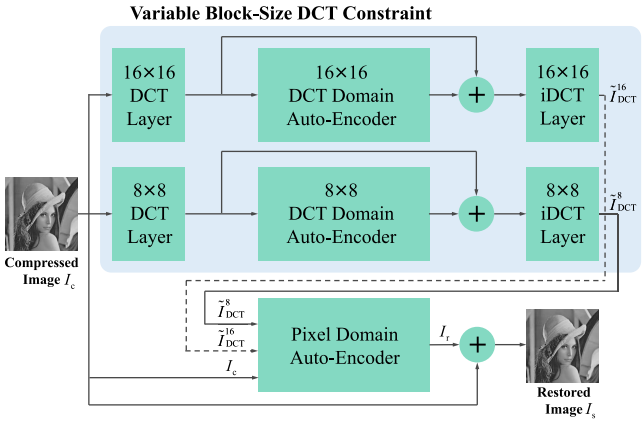
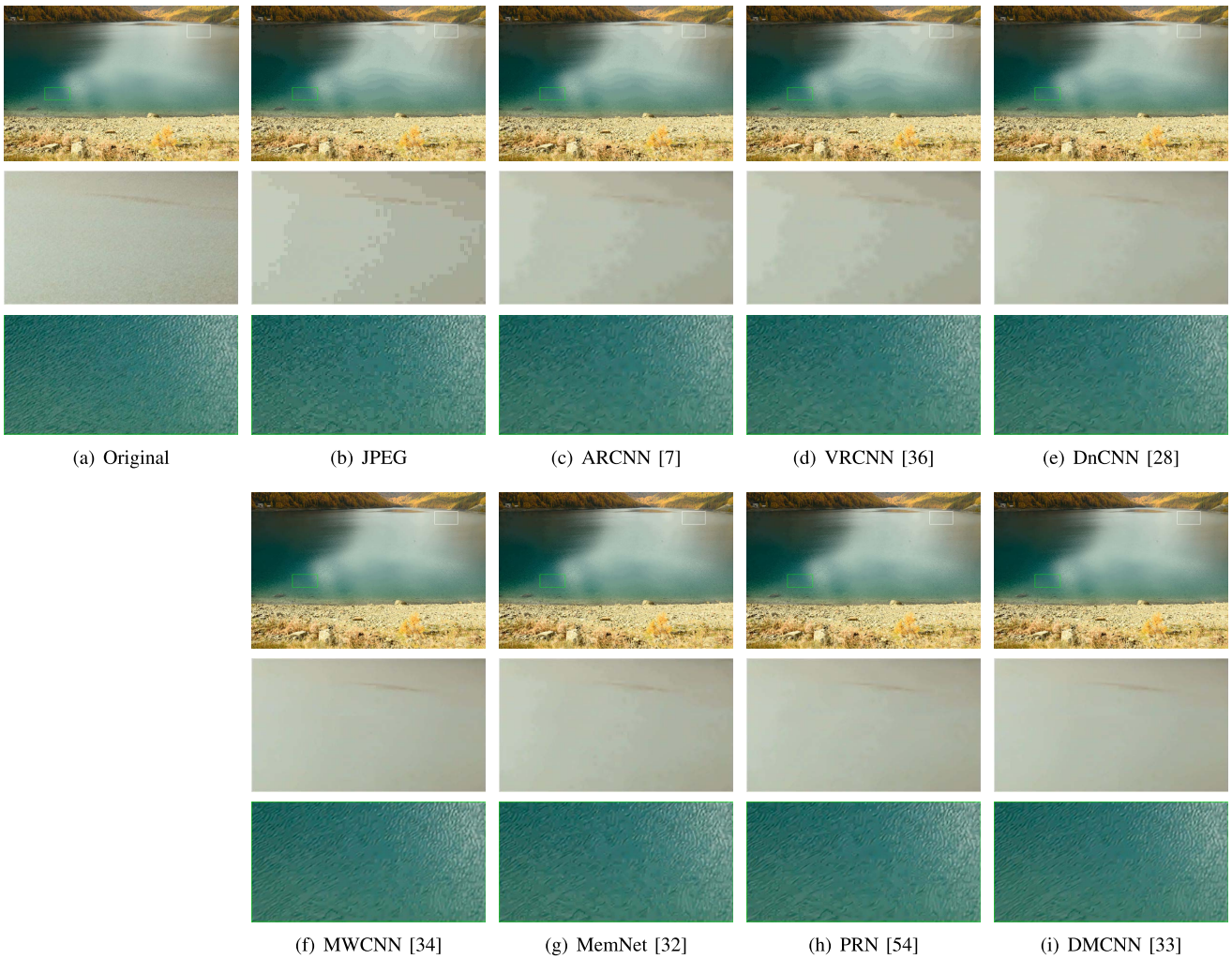



Fig. 6. Variable block-size DCT domain constraint. One DCT branch transforms $8 \times 8$ patches into the DCT domain. While the other DCT branch does the same thing to $16 \times 16$ patches.



(a) Original (b) JPEG (c) ARCNN [7] (d) VRCNN [36] (e) DnCNN [28]

(f) MWCNN [34] (g) MemNet [32] (h) PRN [54] (i) DMCNN [33]

Fig. 7. Examples of restored results for a compressed image utilizing HM from LIU4K (QF = 10).

### *B. Evaluation Protocols*

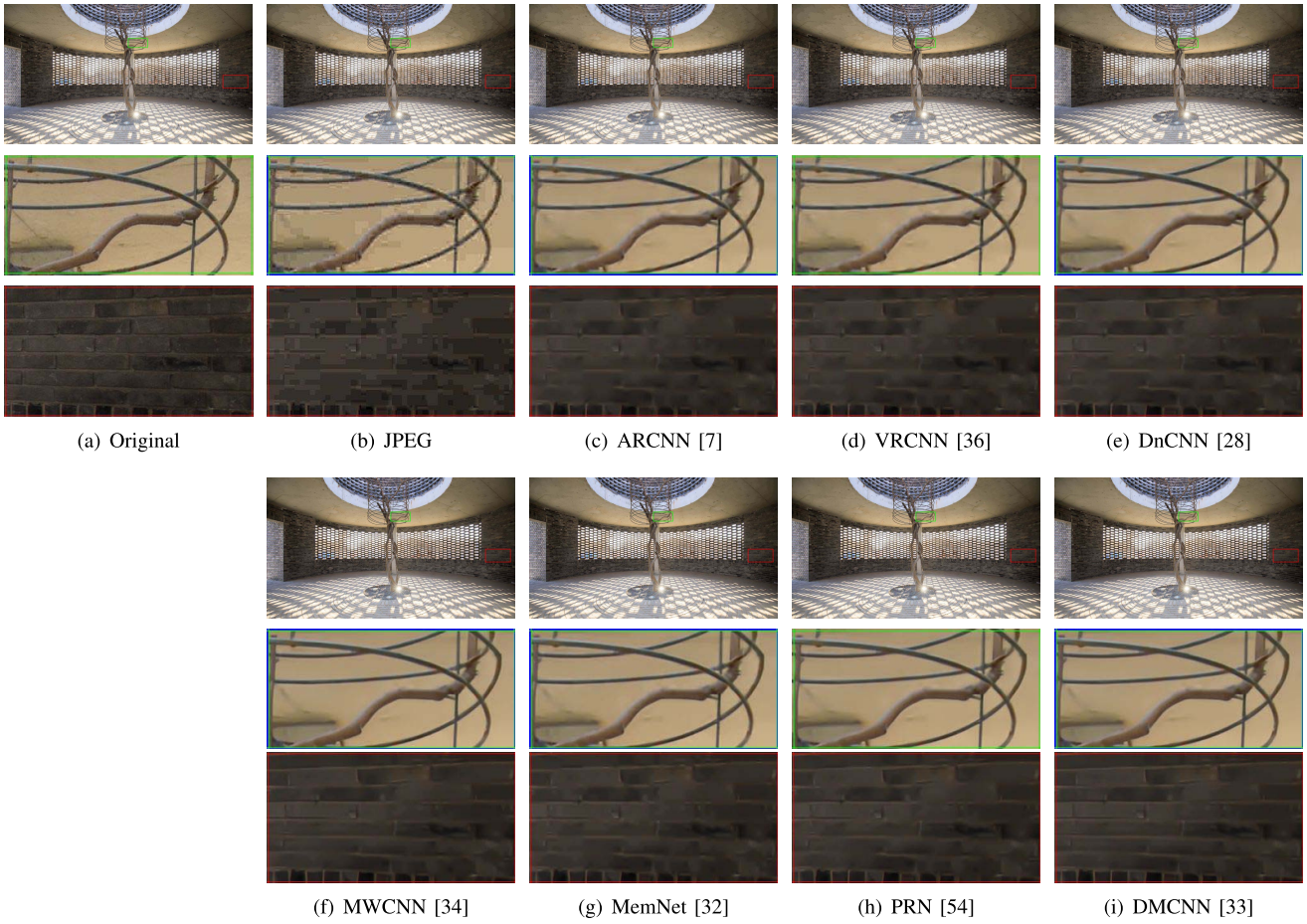
Four full-reference metrics, including PSNR, PSNR-B [59], SSIM [60], MS-SSIM [61], and two non-reference metrics, including NIQE [55], and BRISQUE [62], are used to evaluate the effectiveness of the proposed method. In our implementation, we use the Adam [63] optimizer to pre-train our network and finetune it with stochastic gradient descend (SGD) [64] and cosine decay. In the first stage, the learning rate is set to 0.001. For PRN and MemNet, the learning rate is set to 0.0001. After training 16 epochs, SGD is used for fine-tuning. The initial learning rate is set to 0.0001 at the second-stage of training with cosine decay. We allow at most 60 epochs for JPEG artifact removal and 300 epochs for restoration of compressed images by HEVC. For all methods, the models used for restoring images compressed by JPEG with a quality factor of 40 and HEVC with a quantization parameter of 22 are trained from scratch. Other models are initialized by these two models during the training.

### *C. Objective Comparisons*

The objective results are presented in Table VI. DMCNN is the obvious winner for full-reference metrics, followed by MWCNN for JPEG artifact removal, and PRN for loop filters. On the whole, deep learning-based methods perform significantly better than earlier methods. In no-reference metrics, TNRD achieves a superior performance for JPEG artifact removal and almost all methods generate results that are worse than the original compressed images. We also provide more objective results using different methods on other testing sets in the supplementary materials. These results have high consensus levels among different testing sets.

### *D. Subjective Evaluations*

We also compare the subjective qualities of different methods in Fig. 7 and 8. It is observed that DMCNN achieves the overall best visual quality; most artifacts are removed and texture details are preserved due to the superior modeling capacity. As shown in Fig. 7, JPEG, ARCNN, VRCNN, and DnCNN generate obvious banding effects in large and smooth regions. MemNet and PRN achieve better results. However, one may still discover gentle bands when taking a close look. Benefiting from a large receptive field, MWCNN and DMCNN successfully restore artifacts in the smooth regions and remove banding artifacts. For water wave textures, after compression, some regions are quantized into small smooth blocks. Overall, the methods fail to restore visually pleasing



(a) Original (b) JPEG (c) ARCNN [7] (d) VRCNN [36] (e) DnCNN [28]

(f) MWCNN [34] (g) MemNet [32] (h) PRN [54] (i) DMCNN [33]

Fig. 8. Examples of restored results on a compressed image by JPEG from LIU4K (QF = 10).

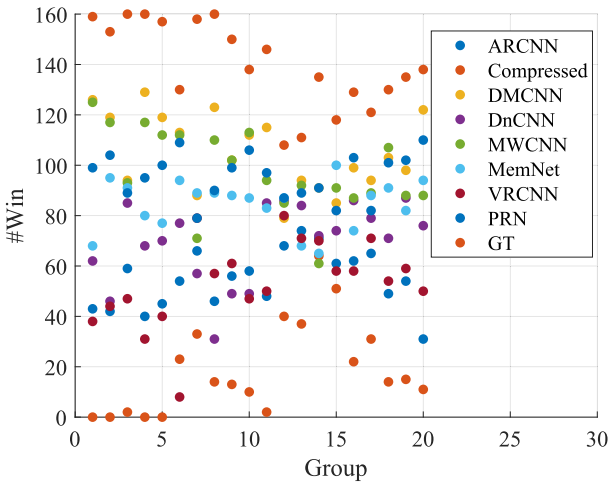



Fig. 9. Visualization of all paired comparisons. The horizontal axis denotes the comparison group ID, while the vertical axis indicates the winning time in the comparison.

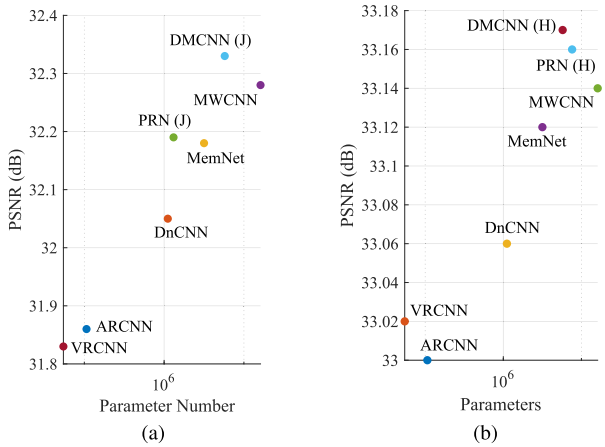



Fig. 10. Visual results of performance and complexity (*i.e.* parameters) of different methods. (a) JPEG artifacts removal (QF = 10). (b) Restoration of compressed images by HEVC (QP = 37).

textures. ARCNN, VRCNN, and DnCNN only remove blockiness boundaries. MemNet and PRN restore water wave textures in stochastic directions. MWCNN and DMCNN generate water wave textures that are consistent with the surrounding waves. Fig. 8 provides the results of edges and regular textures. It is observed that, the results of ARCNN, VRCNN, and DnCNN contain many artifacts. MWCNN, MemNet, and PRN generate better results. DMCNN generates most shape edges and regular brick textures.

We also evaluate the subjective visual quality of different methods using the Mean Opinion Score (MOS) for subjective evaluation. Twenty images are selected from LIVE1, BSD500, Classic5, and LIU4K for the evaluation. These images are compressed by JPEG and HEVC codecs and then processed by different restoration methods. Their results are evaluated

TABLE VI

OBJECTIVE EVALUATIONS OF DIFFERENT METHODS ON LIU4K FOR COMPRESSION ARTIFACT REDUCTION. THE FIRST AND SECOND BEST RESULTS ARE DENOTED IN BOLD AND WITH UNDERLINE, RESPECTIVELY

| Method | Quality | Compressed | SA-DCT | TNRD | ARCNN | VRCNN | DnCNN | DDCN | OTM | MemNet | MWCNN | PRN | DMCNN |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PSNR | QF=10 | 30.45 | 31.32 | 31.85 | 31.86 | 31.83 | 32.05 | 32.09 | 32.18 | 32.18 | 32.28 | 32.19 | **32.33** |
| PSNR-B | | 29.93 | 31.32 | 31.82 | 31.83 | 31.81 | 32.01 | 32.05 | 32.14 | 32.10 | 32.23 | 32.15 | **32.30** |
| SSIM | | 0.8090 | 0.8237 | 0.8427 | 0.8423 | 0.8419 | 0.8463 | 0.8469 | 0.8488 | 0.8488 | 0.8508 | 0.8491 | **0.8520** |
| MS-SSIM | | 0.9270 | 0.9353 | 0.9457 | 0.9457 | 0.9452 | 0.9481 | 0.9488 | 0.9495 | 0.9496 | 0.9511 | 0.9498 | **0.9513** |
| NIQE | | 6.81 | **4.55** | 5.31 | 5.22 | 5.31 | 5.16 | 5.02 | 5.20 | 5.27 | 5.39 | 5.25 | 5.48 |
| BRISQUE | | 56.39 | 51.09 | 43.00 | 49.94 | 48.63 | 49.40 | **40.40** | 41.88 | 49.56 | 50.32 | 50.51 | 51.16 |
| PSNR | QF=20 | 33.28 | 33.87 | 34.48 | 34.51 | 34.47 | 34.57 | 34.73 | 34.78 | 34.80 | 34.86 | 34.83 | **34.91** |
| PSNR-B | | 32.61 | 33.86 | 34.42 | 34.45 | 34.39 | 34.47 | 34.66 | 34.70 | 34.71 | 34.80 | 34.78 | **34.86** |
| SSIM | | 0.8772 | 0.8787 | 0.8958 | 0.8963 | 0.8958 | 0.8970 | 0.8991 | 0.8997 | 0.9005 | 0.9013 | 0.9007 | **0.9023** |
| MS-SSIM | | 0.9675 | 0.9665 | 0.9737 | 0.9741 | 0.9738 | 0.9740 | 0.9751 | 0.9753 | 0.9757 | 0.9760 | 0.9757 | **0.9762** |
| NIQE | | 5.30 | **4.23** | 4.84 | 4.84 | 4.98 | 4.86 | 4.74 | 4.87 | 4.94 | 4.96 | 4.94 | 5.17 |
| BRISQUE | | 53.67 | 46.99 | **39.44** | 45.69 | 47.01 | 43.65 | 41.02 | 42.09 | 46.62 | 45.85 | 46.20 | 47.10 |
| PSNR | QF=30 | 34.81 | 35.27 | 35.92 | 35.90 | 35.89 | 36.11 | 36.20 | 36.24 | 36.21 | 36.23 | 36.23 | **36.31** |
| PSNR-B | | 34.09 | 35.26 | 35.84 | 35.82 | 35.76 | 35.97 | 36.09 | 36.16 | 36.04 | 36.17 | 36.16 | **36.25** |
| SSIM | | 0.9062 | 0.9040 | 0.9192 | 0.9196 | 0.9198 | 0.9215 | 0.9225 | 0.9226 | 0.9229 | 0.9232 | 0.9233 | **0.9242** |
| MS-SSIM | | 0.9799 | 0.9774 | 0.9830 | 0.9832 | 0.9833 | 0.9839 | 0.9841 | 0.9841 | 0.9842 | 0.9844 | 0.9843 | **0.9846** |
| NIQE | | 4.68 | **4.08** | 4.57 | 4.64 | 4.59 | 4.57 | 4.67 | 4.69 | 4.74 | 4.92 | 4.78 | 4.96 |
| BRISQUE | | 48.48 | 45.28 | **37.59** | 43.10 | 42.64 | 42.49 | 40.69 | 41.62 | 43.69 | 44.11 | 43.94 | 44.28 |
| PSNR | QF=40 | 35.82 | 36.20 | - | 36.89 | 36.86 | 37.08 | 37.16 | 37.17 | 37.11 | 37.11 | 37.18 | **37.23** |
| PSNR-B | | 35.07 | 36.19 | - | 36.77 | 36.74 | 36.96 | 37.04 | 37.04 | 36.94 | 37.03 | 37.06 | **37.13** |
| SSIM | | 0.9220 | 0.9188 | - | 0.9331 | 0.9330 | 0.9349 | 0.9350 | 0.9351 | 0.9354 | 0.9353 | 0.9358 | **0.9363** |
| MS-SSIM | | 0.9856 | 0.9829 | - | 0.9878 | 0.9878 | 0.9882 | 0.9882 | 0.9883 | 0.9883 | 0.9884 | 0.9885 | **0.9886** |
| NIQE | | **4.28** | 4.00 | - | 4.49 | 4.54 | 4.63 | 4.59 | 4.54 | 4.61 | 4.75 | 4.68 | 4.80 |
| BRISQUE | | 43.77 | 44.01 | - | 41.35 | 41.00 | 40.79 | **39.71** | 39.75 | 41.28 | 41.51 | 41.56 | 41.80 |
| PSNR | QP=22 | **41.94** | - | - | 41.72 | 41.72 | 41.79 | 41.75 | 41.76 | 41.80 | 41.79 | 41.75 | 41.86 |
| PSNR-B | | 41.67 | - | - | 41.58 | 41.58 | 41.69 | 41.56 | 41.58 | 41.67 | 41.68 | 41.62 | **41.77** |
| SSIM | | 0.9728 | - | - | 0.9729 | 0.9730 | 0.9732 | 0.9727 | 0.9727 | 0.9732 | 0.9732 | 0.9729 | **0.9734** |
| MS-SSIM | | 0.9957 | - | - | 0.9957 | 0.9957 | 0.9957 | 0.9957 | 0.9957 | 0.9957 | 0.9957 | 0.9957 | **0.9958** |
| NIQE | | 3.86 | - | - | 3.80 | 3.81 | 3.93 | **3.64** | 3.66 | 3.96 | 3.91 | 3.78 | 3.97 |
| BRISQUE | | **21.29** | - | - | 24.61 | 24.51 | 24.36 | 31.60 | 31.57 | 24.56 | 24.33 | 23.30 | 24.57 |
| PSNR | QP=27 | 38.47 | - | - | 38.48 | 38.49 | 38.50 | 38.46 | 38.46 | 38.55 | 38.56 | **38.61** | 38.59 |
| PSNR-B | | 38.33 | - | - | 38.43 | 38.45 | 38.47 | 38.40 | 38.39 | 38.50 | 38.52 | 38.56 | **38.57** |
| SSIM | | 0.9456 | - | - | 0.9462 | 0.9462 | 0.9465 | 0.9457 | 0.9456 | 0.9465 | 0.9468 | **0.9473** | 0.9472 |
| MS-SSIM | | 0.9897 | - | - | 0.9896 | 0.9896 | 0.9898 | 0.9896 | 0.9896 | 0.9897 | 0.9898 | **0.9899** | **0.9899** |
| NIQE | | 4.02 | - | - | 4.06 | 4.13 | 4.13 | 3.90 | **3.85** | 4.22 | 4.17 | 4.19 | 4.26 |
| BRISQUE | | **32.68** | - | - | 34.65 | 34.81 | 34.83 | 33.43 | 34.09 | 35.37 | 35.25 | 35.29 | 35.60 |
| PSNR | QP=32 | 35.52 | - | - | 35.63 | 35.65 | 35.71 | 35.60 | 35.61 | 35.74 | 35.75 | 35.78 | **35.79** |
| PSNR-B | | 35.46 | - | - | 35.62 | 35.64 | 35.70 | 35.58 | 35.60 | 35.71 | 35.74 | 35.77 | **35.78** |
| SSIM | | 0.9061 | - | - | 0.9072 | 0.9073 | 0.9083 | 0.9065 | 0.9068 | 0.9082 | 0.9085 | 0.9091 | **0.9094** |
| MS-SSIM | | 0.9776 | - | - | 0.9777 | 0.9776 | 0.9780 | 0.9776 | 0.9777 | 0.9779 | 0.9781 | 0.9782 | **0.9783** |
| NIQE | | 4.55 | - | - | 4.57 | 4.62 | 4.68 | **4.29** | 4.33 | 4.76 | 4.61 | 4.74 | 4.79 |
| BRISQUE | | 40.99 | - | - | 41.87 | 42.58 | 42.47 | 35.98 | **35.41** | 42.61 | 43.32 | 43.51 | 43.69 |
| PSNR | QP=37 | 32.85 | - | - | 33.00 | 33.02 | 33.06 | 32.98 | 32.98 | 33.12 | 33.14 | 33.16 | **33.17** |
| PSNR-B | | 32.81 | - | - | 33.00 | 33.01 | 33.05 | 32.97 | 32.97 | 33.10 | 33.14 | 33.14 | **33.16** |
| SSIM | | 0.8558 | - | - | 0.8577 | 0.8582 | 0.8582 | 0.8571 | 0.8570 | 0.8594 | 0.8604 | 0.8603 | **0.8613** |
| MS-SSIM | | 0.9559 | - | - | 0.9563 | 0.9562 | 0.9564 | 0.9560 | 0.9559 | 0.9567 | 0.9573 | 0.9571 | **0.9575** |
| NIQE | | 5.04 | - | - | 5.03 | 5.07 | 5.07 | **4.65** | 4.74 | 5.24 | 5.08 | 5.14 | 5.21 |
| BRISQUE | | 46.61 | - | - | 47.16 | 47.62 | 46.83 | 38.11 | **36.33** | 48.74 | 47.77 | 47.11 | 48.30 |

TABLE VII

THE MODEL COMPLEXITY ANALYSIS OF DIFFERENT METHODS. **J** DENOTES THE VERSION USED FOR JPEG ARTIFACTS REMOVAL. **H** SIGNIFIES THE VERSION USED FOR THE RESTORATION OF COMPRESSED IMAGES BY HEVC

| Categories | Non-Deep | Deep JPEG Artifacts Removal | | | | | | | Deep Loop Filter | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Metrics | SA-DCT | ARCNN | TNRD | DnCNN | MemNet | MWCNN | PRN (J) | DMCNN (J) | VRCNN | PRN (H) | DMCNN (H) |
| Parameter | - | 106,564 | 26,645 | 1,112,192 | 3,165,196 | 16,152,260 | 1,312,140 | 5,751,614 | 54,673 | 7,600,065 | 9,400,180 |
| Storage (MB) | - | 0.40 | 1.93 | 2.12 | 12.26 | 61.60 | 5.16 | 21.95 | 0.21 | 29.16 | 22.67 |
| Time (ms/per-image) | 43164.00 | 3.56 | 15050.30 | 6.31 | 186.30 | 132.39 | 49.92 | 17.34 | 3.92 | 841.01 | 32.48 |

by human annotators. Forty participants are invited to join the subjective experiment. Each individual is required to provide subjective results for 360 image pairs. The comparison results are illustrated in Fig. 9 and Table VIII. Based on the compared pairs, we also fit a Bradley-Terry model [77] to estimate the MOS score for each method so that they can be ranked. The inferred average MOS score is presented in Table IX. It is observed that, DMCNN, MemNet, and PRN achieve overall superior visual quality than other methods.

### *E. Evaluation of Model Capacity*

Table VII reports the parameter number, the storage usage, and the per-image running time for each method averaged over images ($768 \times 512$) from LIVE1 on a machine with Intel(R) Xeon(TM) E5-2650 v4 2.20 GHz CPU, 16G RAM, and GeForce GTX 1080 Ti. ARCNN, DnCNN, MemNet, MWCNN, PRN, DMCNN, and VRCNN are implemented in Pytorch. SA-DCT and TNRD are implemented in MATLAB. ARCNN, DnCNN, MemNet, MWCNN, PRN, DMCNN, and VRCNN run on GPU while SA-DCT and TNRD run on CPU. It is observed that all deep learning-based methods finish processing an image within one second. ARCNN, VRCNN and DnCNN achieve the shortest running times and finish the restoration within ten milliseconds. As for storage, ARCNN and VRCNN use the least storage space. As for model complexity, MWCNN uses the most parameters while TNRD uses the fewest. Note that PRN and DMCNN use different network architectures to handle JPEG artifact removal

TABLE VIII

THE RESULTS OF PAIRWISE COMPARISON IN A USER STUDY. EACH VALUE REPRESENTS THE NUMBER OF TIMES THE METHOD IN EACH ROW HAS OUTPERFORMED THE METHOD IN THE RESPECTIVE COLUMN

| | Compressed | ARCNN | DMCNN | DnCNN | MWCNN | MemNet | VRCNN | PRN | GT |
|---|---|---|---|---|---|---|---|---|---|
| Compressed | - | 57 | 51 | 57 | 53 | 56 | 64 | 24 | 48 |
| ARCNN | 370 | - | 95 | 149 | 79 | 99 | 224 | 43 | 92 |
| DMCNN | 376 | 332 | - | 315 | 266 | 284 | 343 | 61 | 275 |
| DnCNN | 370 | 278 | 112 | - | 118 | 161 | 276 | 57 | 115 |
| MWCNN | 374 | 348 | 161 | 309 | - | 253 | 335 | 68 | 223 |
| MemNet | 371 | 328 | 143 | 266 | 174 | - | 319 | 61 | 153 |
| VRCNN | 363 | 203 | 84 | 151 | 92 | 108 | - | 44 | 95 |
| PRN | 379 | 335 | 152 | 312 | 204 | 274 | 332 | - | 65 |
| GT | 403 | 384 | 366 | 370 | 359 | 366 | 383 | 362 | - |

TABLE IX

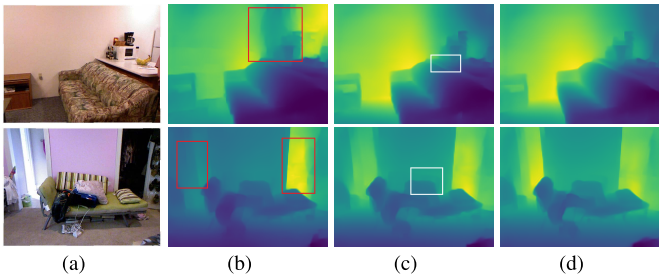
THE MOS SCORE OF DIFFERENT METHODS

| Compressed | ARCNN | DMCNN | DnCNN | MWCNN | MemNet | VRCNN | PRN | GT |
|---|---|---|---|---|---|---|---|---|
| 0.330 | 0.094 | 1.371 | 0.515 | 1.082 | 0.781 | 0.325 | 1.000 | 3.753 |

and the restoration of compressed images by HEVC. Therefore, we present the complexities of the different versions in Table VII, depicted by (J) and (H), respectively. The results are also depicted in Fig. 10.

### F. Performance Evaluations of Computer Vision Tasks

*1) Depth Estimation:* Table XI shows the results of depth estimation with accurate object boundaries [67]. This is one of the state-of-the-art depth estimation method, based on images with and without compression artifact reduction by VRCNN on NYUv2 [68] in different measures. Several accuracy measures are employed to evaluate the depth estimation performance: mean squared error (MSE), root mean squared error (RMS), mean relative error (MRE), mean log 10 error (log 10), and threshold accuracy, as well as precision (P), recall (R), and F1 score of estimated edge maps. It is noteworthy that for MSE, RMS, and MRE, small values signify better performance. For log 10, threshold accuracy $\delta$, P, R, and F1 score, large values denote better performance. Judging from the results, for MSE, RMS, and MRE, it is always beneficial to perform compression artifact reduction among all cases (both JPEG and HEVC codecs with all QPs and QFs); whereas, for other metrics, the results become slightly controversial. The results of the restored images are sometimes inferior to those of the compressed ones, *e.g.*, QF $=$ 10 on log 10, and QF $=$ 30, 40 on P, *etc*. However, in general, the results of restored images are successful in more cases compared to compressed ones. It is also demonstrated that a reconstruction aiming to restore compressed images with high visual quality might not always be beneficial for all tasks. The trend of performance changes taking place before and after restoration at different QP/QF conditions is also illustrated in Fig. 12, and visual results are shown in Fig. 11. It is observed that when QF $=$ 10, the result of a compressed image degrades extensively, and the enhancement operation effectively improves the visual quality of depth maps. When QF $=$ 20, the degradations in the result of a compressed image are not obvious. Enhancement operations also lead to minor visual quality gains. Some discontinuous boundary artifacts are removed, as shown in the red boxes in Fig 11. However, some details become still blurry, *e.g.* the details and boundaries in the white boxes, as shown in Fig 11.



(a) (b) (c) (d)

Fig. 11. The visual results of depth estimation on compressed images (JPEG) with and without compression artifact reduction. (a) Input RGB image. (b) Depth map of compressed image (QF $=$ 10). (c) Depth map of restored image (QF $=$ 10). (d) Depth map of original image.

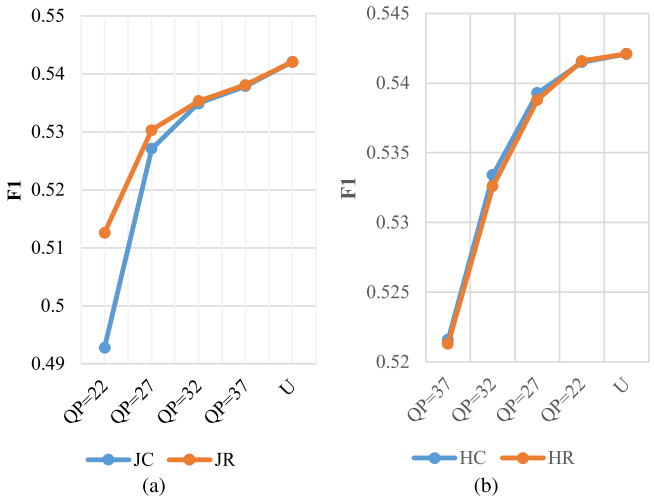



Fig. 12. Visual results of performance changes before and after restorations at different QP/QF conditions for depth estimation. JC: Compressed by JPEG. HC: Compressed by HEVC. JR: Restored from images compressed by JPEG. HR: Restored from images compressed images by HEVC.

*2) Semantic Segmentation:* We integrate two baselines for evaluations: ResNet50Dilated + PPM_Deepsup and ResNet50 + UperNet [65]. Evaluations are performed on ADE20K [65]. Results are reported in two metrics commonly used for semantic segmentation [66]: Pixel accuracy

TABLE X

COMPARISONS OF SENET ON THE NYU-DEPTH V2 DATASET. INPUT TESTING IMAGES IN THE "COMPRESSED" CATEGORY ARE COMPRESSED WITH DIFFERENT QUANTIZATION PARAMETERS. INPUT TESTING IMAGES IN THE "RESTORED" CATEGORY ARE PROCESSED WITH VRCNN. "QF" DENOTES THE QUALITY FACTOR. "QP" SIGNIFIES THE QUANTIZATION PARAMETER

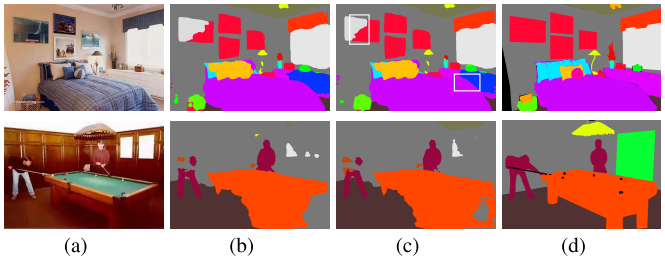
| Metrics | Codecs | Uncompressed | Compressed | | | | Restored | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| QF | - | - | QF=10 | QF=20 | QF=30 | QF=40 | QF=10 | QF=20 | QF=30 | QF=40 |
| MSE | JPEG | 0.2994 | 0.4072 | 0.3193 | 0.3080 | 0.3031 | 0.3652 | 0.3163 | 0.3058 | 0.3021 |
| RMS | | 0.5471 | 0.6382 | 0.5651 | 0.5550 | 0.5505 | 0.6043 | 0.5624 | 0.5530 | 0.5496 |
| REL | | 0.1179 | 0.1482 | 0.1248 | 0.1207 | 0.1191 | 0.1354 | 0.1245 | 0.1214 | 0.1201 |
| log10 | | 0.0521 | 0.0644 | 0.0548 | 0.0531 | 0.0526 | 0.0611 | 0.0551 | 0.0535 | 0.0530 |
| $\delta < 1.25$ | | 0.8572 | 0.7910 | 0.8414 | 0.8529 | 0.8546 | 0.8057 | 0.8384 | 0.8488 | 0.8520 |
| $\delta < 1.25$ | | 0.9728 | 0.9484 | 0.9689 | 0.9711 | 0.9720 | 0.9531 | 0.9679 | 0.9703 | 0.9715 |
| $\delta < 1.25^3$ | | 0.9928 | 0.9864 | 0.9914 | 0.9921 | 0.9927 | 0.9888 | 0.9917 | 0.9922 | 0.9926 |
| P | | 0.6220 | 0.5694 | 0.6091 | 0.6167 | 0.6193 | 0.6020 | 0.6099 | 0.6133 | 0.6156 |
| R | | 0.4904 | 0.4481 | 0.4751 | 0.4825 | 0.4853 | 0.4575 | 0.4795 | 0.4851 | 0.4880 |
| F1 | | 0.5421 | 0.4928 | 0.5271 | 0.5349 | 0.5379 | 0.5126 | 0.5303 | 0.5353 | 0.5381 |
| QP | - | - | QP=22 | QP=27 | QP=32 | QP=37 | QP=22 | QP=27 | QP=32 | QP=37 |
| MSE | HEVC | 0.2994 | 0.2976 | 0.2973 | 0.3129 | 0.3385 | 0.2971 | 0.2991 | 0.3183 | 0.3483 |
| RMS | | 0.5471 | 0.5455 | 0.5453 | 0.5594 | 0.5818 | 0.5451 | 0.5469 | 0.5642 | 0.5901 |
| REL | | 0.1179 | 0.1185 | 0.1199 | 0.1235 | 0.1300 | 0.1188 | 0.1204 | 0.1246 | 0.1319 |
| log10 | | 0.0521 | 0.0522 | 0.0527 | 0.0546 | 0.0580 | 0.0523 | 0.0530 | 0.0553 | 0.0594 |
| $\delta < 1.25$ | | 0.8572 | 0.8553 | 0.8535 | 0.8420 | 0.8247 | 0.8547 | 0.8517 | 0.8376 | 0.8171 |
| $\delta < 1.25^2$ | | 0.9728 | 0.9725 | 0.9714 | 0.9682 | 0.9617 | 0.9724 | 0.9709 | 0.9664 | 0.9581 |
| $\delta < 1.25^3$ | | 0.9928 | 0.9929 | 0.9926 | 0.9918 | 0.9900 | 0.9930 | 0.9924 | 0.9916 | 0.9895 |
| P | | 0.6220 | 0.6196 | 0.6133 | 0.6090 | 0.6030 | 0.6188 | 0.6130 | 0.6098 | 0.6065 |
| R | | 0.4904 | 0.4909 | 0.4913 | 0.4847 | 0.4710 | 0.4915 | 0.4909 | 0.4833 | 0.4687 |
| F1 | | 0.5421 | 0.5415 | 0.5393 | 0.5334 | 0.5216 | 0.5416 | 0.5388 | 0.5326 | 0.5213 |

TABLE XI

COMPARISONS OF SENET ON THE NYU-DEPTH V2 DATASET. INPUT TESTING IMAGES IN THE "COMPRESSED" CATEGORY ARE COMPRESSED WITH DIFFERENT QUANTIZATION PARAMETERS. INPUT TESTING IMAGES IN THE "PREDICTED" CATEGORY ARE PROCESSED WITH VRCNN. "QF" DENOTES THE QUALITY FACTOR. "QP" SIGNIFIES THE QUANTIZATION PARAMETER

| Metrics | Codecs | Baseline | Uncompressed | Compressed | | | | Restored | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| QF | - | - | - | QF=10 | QF=20 | QF=30 | QF=40 | QF=10 | QF=20 | QF=30 | QF=40 |
| Mean IoU | JPEG | ResNet50Dilated + PPM_Deepsup | 0.4075 | 0.2794 | 0.3711 | 0.3911 | 0.394 | 0.3145 | 0.3624 | 0.382 | 0.3831 |
| Accuracy | | | 79.64% | 70.77% | 77.17% | 78.49% | 78.78% | 73.98% | 77.15% | 78.26% | 78.41% |
| Mean IoU | | ResNet50 + UperNet | 0.4029 | 0.2814 | 0.3662 | 0.3842 | 0.3884 | 0.3345 | 0.3685 | 0.3834 | 0.3843 |
| Accuracy | | | 79.57% | 70.85% | 77.32% | 78.47% | 78.65% | 75.40% | 77.56% | 78.48% | 78.45% |
| QP | - | - | - | QP=22 | QP=27 | QP=32 | QP=37 | QP=22 | QP=27 | QP=32 | QP=37 |
| Mean IoU | HEVC | ResNet50Dilated + PPM_Deepsup | 0.4075 | 0.4043 | 0.3956 | 0.3736 | 0.3363 | 0.4027 | 0.3902 | 0.3653 | 0.3269 |
| Accuracy | | | 79.64% | 79.50% | 79.02% | 77.75% | 75.08% | 79.45% | 78.80% | 77.41% | 74.66% |
| Mean IoU | | ResNet50 + UperNet | 0.4029 | 0.4006 | 0.3945 | 0.3789 | 0.3463 | 0.3999 | 0.3931 | 0.3744 | 0.3417 |
| Accuracy | | | 79.57% | 79.45% | 79.03% | 78.12% | 76.02% | 79.42% | 78.94% | 77.95% | 75.84% |

indicates the proportion of correctly classified pixels. Mean IoU indicates the intersection-over-union between the predicted and groundtruth pixel, averaged over all the classes. It is observed from Table XI that compression artifact reduction (*i.e.* VRCNN) may not benefit the inference of semantic segmentation all the time. In many cases, *e.g.* for JPEG artifacts, the performance of the baseline ResNet50Dilated + PPM_Deepsup for restored images is worse than with compressed images when QF = 40. The trend of performance changes occurring before and after the restorations at different QP/QF conditions is also depicted in Fig. 14. The main reason for the performance drop might be the consensus of the effects of MSE used in training and the semantic segmentation purposes. When training with MSE, the restoration results for compressed images with gender artifacts tend to be smooth, and some critical details are lost causing low accuracy. For visual results, it is observed from Fig. 13 that, compression artifact removal slightly corrects some false boundaries.



(a) (b) (c) (d)

Fig. 13. The visual results of semantic segmentation on compressed images (HEVC) with and without compression artifact reduction. (a) Input RGB image. (b) Semantic map of compressed image (QP = 37). (c) Semantic map of restored image (QP = 37). (d) Semantic map of original image.

## V. TRENDS AND CHALLENGES

Although deep learning techniques for compression artifact reduction have developed rapidly, several important challenges and inherent patterns remain. First, recent researchers have obtained higher and higher accuracy by using advanced deep models with a huge amount of parameters; however, it is still hard to apply these methods in real scenarios. It is interesting to re-design compact deep network architectures and compress or adjust the existing models into small ones for *real-time compression artifact reduction*. Second, with the latest codecs, *i.e.* versatile video coding (VVC), more integrated tools are

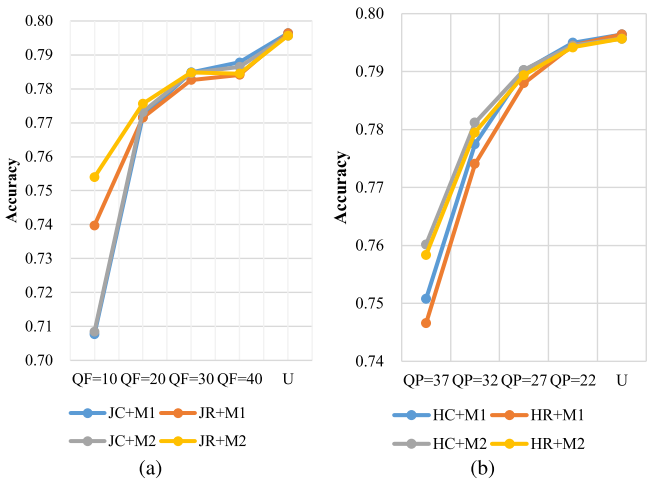



Fig. 14. Visual results of performance changes before and after the restoration at different QP/QF conditions for semantic segmentation. JC: Compressed by JPEG. HC: Compressed by HEVC. JR: Restored from images compressed by JPEG. HR: Restored from images compressed images by HEVC. M1: ResNet50Dilated + PPM Deepsup. M2: ResNet50 + UperNet.

employed, thus the distribution of compression artifacts is more complex. It is challenging to apply the existing methods to the next generation of codecs. With more powerful tools for deep learning, *e.g.* capsule networks, and reinforcement learning *etc.*, we believe that, the future *technique improvement on restoration of more complex degradations* will yield new surprises. Third, for compression artifacts reduction, there are few works on the *internal mechanism of feature learning and related interpretable factors*. Beyond obtaining superior performance, one future direction is to give comprehensive explanations of what factors lead to a more effective network and a more specific mechanism. Finally, for various low-level image processing tasks, it is critical to design and apply proper metrics to constrain model training and evaluate a model's effectiveness. Thus, it is an important future goal to develop more effective and rational measures that balance both signal fidelity and visual perception for compression artifact reduction.

## VI. Conclusion

This paper presents a systematic review of compression artifact reduction methods, including both traditional and deep-learning based methods. These methods have evolved from several perspectives, including model architecture improvement and continuing exploration of side information embedding, *etc*. We summarize milestones and typical methods and highlight their contributions, strengths, and weaknesses. We also create a thorough benchmark for state-of-the-art compression artifact reduction methods. In our benchmarking experiments, some constraints and training skills targeted for JPEG artifact removal are generalized to handle general compression artifacts reduction methods. Based on our evaluation and analysis, overall remarks, challenges, and trends are given. Although our attempts are preliminary, they build a bridge from the existing world to a new one, where more researchers are expected to come.

## References

[1] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consum. Electron.*, vol. 38, no. 1, pp. xviii–xxxiv, Feb. 1992.

[2] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[3] I. E. Richardson, *The H.264 Advanced Video Compression Standard*, 2nd ed. Hoboken, NJ, USA: Wiley, 2010.

[4] K. Bredies and M. Holler, "A total variation-based JPEG decompression model," *SIAM J. Imag. Sci.*, vol. 5, no. 1, pp. 366–393, Jan. 2012.

[5] K. Lee, D. Sik Kim, and T. Kim, "Regression-based prediction for blocking artifact reduction in JPEG-compressed images," *IEEE Trans. Image Process.*, vol. 14, no. 1, pp. 36–48, Jan. 2005.

[6] H. Chang, M. K. Ng, and T. Zeng, "Reducing artifacts in JPEG decompression via a learned dictionary," *IEEE Trans. Signal Process.*, vol. 62, no. 3, pp. 718–728, Feb. 2014.

[7] C. Dong, Y. Deng, C. C. Loy, and X. Tang, "Compression artifacts reduction by a deep convolutional network," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 576–584.

[8] Z. Wang, D. Liu, S. Chang, Q. Ling, Y. Yang, and T. S. Huang, "D3: Deep dual-domain based fast restoration of JPEG-compressed images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2764–2772.

[9] J. Guo and H. Chao, "Building dual-domain representations for compression artifacts reduction," in *Proc. ECCV*, 2016, pp. 628–644.

[10] Y. Wang, H. Zhu, Y. Li, Z. Chen, and S. Liu, "Dense residual convolutional neural network based in-loop filter for HEVC," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Dec. 2018, pp. 1–4.

[11] X. He, Q. Hu, X. Zhang, C. Zhang, W. Lin, and X. Han, "Enhancing HEVC compressed videos with a partition-masked convolutional neural network," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 216–220.

[12] 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. (ICCV)*, Jul. 2001, pp. 416–423.

[13] R. Timofte, S. Gu, J. Wu, and L. Van Gool, "NTIRE 2018 challenge on single image super-resolution: Methods and results," in *Proc. CVPR*, Jun. 2018, pp. 852–863.

[14] 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.

[15] A. Singh and N. Ahuja, "Super-resolution using sub-band self-similarity," in *Proc. ACCV*, 2015, pp. 552–568.

[16] A. Foi, V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images," *IEEE Trans. Image Process.*, vol. 16, no. 5, pp. 1395–1411, May 2007.

[17] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4681–4690.

[18] S. Minami and A. Zakhor, "An optimization approach for removing blocking effects in transform coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, no. 2, pp. 74–82, Apr. 1995.

[19] C.-Y. Tsai *et al.*, "Adaptive loop filtering for video coding," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 6, pp. 934–945, Dec. 2013.

[20] D. Sun and W.-K. Cham, "Postprocessing of low bit-rate block DCT coded images based on a fields of experts prior," *IEEE Trans. Image Process.*, vol. 16, no. 11, pp. 2743–2751, Nov. 2007.

[21] X. Zhang, R. Xiong, S. Ma, and W. Gao, "Reducing blocking artifacts in compressed images via transform-domain non-local coefficients estimation," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2012, pp. 836–841.

[22] X. Zhang, R. Xiong, X. Fan, S. Ma, and W. Gao, "Compression artifact reduction by overlapped-block transform coefficient estimation with block similarity," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 4613–4626, Dec. 2013.

[23] J. Ren, J. Liu, M. Li, W. Bai, and Z. Guo, "Image blocking artifacts reduction via patch clustering and low-rank minimization," in *Proc. Data Compress. Conf.*, Mar. 2013, p. 516.

[24] X. Liu, X. Wu, J. Zhou, and D. Zhao, "Data-driven sparsity-based restoration of JPEG-compressed images in dual transform-pixel domain," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5171–5178.

[25] X. Liu, X. Wu, J. Zhou, and D. Zhao, "Data-driven soft decoding of compressed images in dual transform-pixel domain," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1649–1659, Apr. 2016.

[26] X. Zhang, W. Lin, R. Xiong, X. Liu, S. Ma, and W. Gao, "Low-rank decomposition-based restoration of compressed images via adaptive noise estimation," *IEEE Trans. Image Process.*, vol. 25, no. 9, pp. 4158–4171, Sep. 2016.

[27] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1256–1272, Jun. 2017.

[28] 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.

[29] X. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *Proc. NIPS*, 2016, pp. 2802–2810.

[30] L. Cavigelli, P. Hager, and L. Benini, "CAS-CNN: A deep convolutional neural network for image compression artifact suppression," in *Proc. IJCNN*, 2017, pp. 752–759.

[31] J. Guo and H. Chao, "One-to-many network for visually pleasing compression artifacts reduction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3038–3047.

[32] Y. Tai, J. Yang, X. Liu, and C. Xu, "MemNet: A persistent memory network for image restoration," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4539–4547.

[33] X. Zhang, W. Yang, Y. Hu, and J. Liu, "DMCNN: Dual-domain multi-scale convolutional neural network for compression artifacts removal," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 390–394.

[34] P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo, "Multi-level wavelet-CNN for image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 773–782.

[35] H. Chen, X. He, L. Qing, S. Xiong, and T. Q. Nguyen, "DPW-SDNet: Dual pixel-wavelet domain deep CNNs for soft decoding of JPEG-compressed images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 711–720.

[36] Y. Dai, D. Liu, and F. Wu, "A convolutional neural network approach for post-processing in HEVC intra coding," in *Proc. MMM*, 2017, pp. 28–39.

[37] T. Wang, M. Chen, and H. Chao, "A novel deep learning-based method of improving coding efficiency from the decoder-end for HEVC," in *Proc. Data Compress. Conf. (DCC)*, Apr. 2017, pp. 410–419.

[38] Y. Zhang, T. Shen, X. Ji, Y. Zhang, R. Xiong, and Q. Dai, "Residual highway convolutional neural networks for in-loop filtering in HEVC," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3827–3841, Aug. 2018.

[39] X. Meng, C. Chen, S. Zhu, and B. Zeng, "A new HEVC in-loop filter based on multi-channel long-short-term dependency residual networks," in *Proc. Data Compress. Conf.*, Mar. 2018, pp. 187–196.

[40] Z. Jin, P. An, C. Yang, and L. Shen, "Quality enhancement for intra frame coding via CNNS: An adversarial approach," in *Proc. ICASSP*, Apr. 2018, pp. 1368–1372.

[41] R. Yang, M. Xu, and Z. Wang, "Decoder-side HEVC quality enhancement with scalable convolutional neural network," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2017, pp. 817–822.

[42] X. Song *et al.*, "A practical convolutional neural network as loop filter for intra frame," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 1133–1137.

[43] T. Wang, W. Xiao, M. Chen, and H. Chao, "The multi-scale deep decoder for the standard HEVC bitstreams," in *Proc. Data Compress. Conf.*, Mar. 2018, pp. 197–206.

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

[45] T. Hashimoto, E. Sasaki, and T. Ika. (Jul. 2018). *JVET-K0158: Separable Convolutional Neural Network Filter With Squeeze-and-Excitation Block*. [Online]. Available: http://phenix.it-sudparis.eu/jvet/

[46] H. C. Reeve, III, and J. S. Lim, "Reduction of blocking effects in image coding," *Opt. Eng.*, vol. 23, no. 1, Feb. 1984, Art. no. 230134.

[47] B. Ramamurthi and A. Gersho, "Nonlinear space-variant postprocessing of block coded images," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 34, no. 5, pp. 1258–1268, Oct. 1986.

[48] J. Jancsary, S. Nowozin, and C. Rother, "Loss-specific training of non-parametric image restoration models: A new state of the art," in *Proc. ECCV*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds., 2012, pp. 112–125.

[49] K. Yu, C. Dong, C. Change Loy, and X. Tang, "Deep convolution networks for compression artifacts reduction," 2016, *arXiv:1608.02778*. [Online]. Available: http://arxiv.org/abs/1608.02778

[50] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. ICML*, 2010, pp. 399–406.

[51] J. Kang, S. Kim, and K. M. Lee, "Multi-modal/multi-scale convolutional neural network based in-loop filter design for next generation video codec," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 26–30.

[52] C. Li, L. Song, R. Xie, and W. Zhang, "CNN based post-processing to improve HEVC," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 4577–4580.

[53] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1646–1654.

[54] D. Wang, S. Xia, W. Yang, Y. Hu, and J. Liu, "Partition tree guided progressive rethinking network for in-loop filtering of HEVC," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 2671–2675.

[55] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a 'completely blind' image quality analyzer," *IEEE Signal Process. Lett.*, vol. 20, no. 3, pp. 209–212, Mar. 2013.

[56] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Trans. Image Process.*, vol. 21, no. 12, pp. 4695–4708, Dec. 2012.

[57] X. Chen, Q. Zhang, M. Lin, G. Yang, and C. He, "No-reference color image quality assessment: From entropy to perceptual quality," *EURASIP J. Image Video Process.*, vol. 2019, Sep. 2019, Art. no. 77.

[58] C. Ma, C.-Y. Yang, X. Yang, and M.-H. Yang, "Learning a no-reference quality metric for single-image super-resolution," *Comput. Vis. Image Understand.*, vol. 158, pp. 1–16, May 2017.

[59] C. Yim and A. C. Bovik, "Quality assessment of deblocked images," *IEEE Trans. Image Process.*, vol. 20, no. 1, pp. 88–98, Jan. 2011.

[60] Z. 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, Apr. 2004.

[61] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. ACSSC*, Nov. 2003, pp. 1398–1402.

[62] A. Mittal, A. K. Moorthy, and A. C. Bovik, "Blind/referenceless image spatial quality evaluator," in *Proc. Conf. Rec. Forty 5th Asilomar Conf. Signals, Syst. Comput. (ASILOMAR)*, Nov. 2011, pp. 723–727.

[63] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2014, pp. 1–15.

[64] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. COMPSTAT*, 2010, pp. 177–186.

[65] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ADE20K dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 633–641.

[66] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.

[67] J. Hu, M. Ozay, Y. Zhang, and T. Okatani, "Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 1043–1051.

[68] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. ECCV*, 2012, pp. 746–760.

[69] A. Norkin *et al.*, "HEVC deblocking filter," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1746–1754, Dec. 2012.

[70] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using MATLAB*. Upper Saddle River, NJ, USA: Prentice-Hall, 2003.

[71] L. Feng, X. Zhang, S. Wang, Y. Wang, and S. Ma, "Coding prior based high efficiency restoration for compressed video," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 769–773.

[72] Y. Dai, D. Liu, Z.-J. Zha, and F. Wu, "A CNN-based in-loop filter with CU classification for HEVC," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Dec. 2018, pp. 1–4.

[73] 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.

[74] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

[75] H. R. Sheikh, M. F. Sabir, and A. C. Bovik, "A statistical evaluation of recent full reference image quality assessment algorithms," *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3440–3451, Nov. 2006.

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

[77] R. A. Bradley and M. E. Terry, "Rank analysis of incomplete block designs: I. The method of paired comparisons," *Biometrika*, vol. 39, pp. 324–345, Dec. 1952.

**Jiaying Liu** (Senior Member, IEEE) received the Ph.D. degree (Hons.) in computer science from Peking University, Beijing China, in 2010.

She was a Visiting Scholar with the University of Southern California, Los Angeles, from 2007 to 2008. She was a Visiting Researcher with Microsoft Research Asia in 2015 supported by the Star Track Young Faculties Award. She is currently an Associate Professor with the Wangxuan Institute of Computer Technology, Peking University. She has authored over 100 technical papers in refereed journals and proceedings and holds 43 granted patents. Her current research interests include multimedia signal processing, compression, and computer vision. She is a Senior Member of CSIG and CCF. She has served as a member of the Membership Services Committee in the IEEE Signal Processing Society, the Multimedia Systems and Applications Technical Committee (MSA TC), the Visual Signal Processing and Communications Technical Committee (VSPC TC) in the IEEE Circuits and Systems Society, and the Image, Video, and Multimedia (IVM) Technical Committee in APSIPA. She has also served as an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING and JVCI (Elsevier), the Technical Program Chair of the IEEE ICME-2021/VCIP-2019 and ACM ICMR-2021, and the Area Chair of CVPR-2021/ECCV-2020/ICCV-2019. She was an APSIPA Distinguished Lecturer from 2016 to 2017.

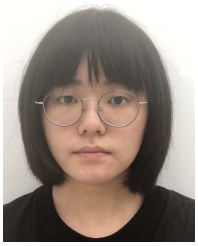
**Dong Liu** (Senior Member, IEEE) received the B.S. and Ph.D. degrees in electrical engineering from the University of Science and Technology of China (USTC), Hefei, China, in 2004 and 2009, respectively.

He was a Member of Research Staff with Nokia Research Center, Beijing, China, from 2009 to 2012. He joined USTC as an Associate Professor in 2012. His research interests include image and video coding, multimedia signal processing, and multimedia data mining. He has authored or coauthored more than 100 papers in international journals and conferences. He has 16 granted patents. He has several technical proposals adopted by international and domestic standardization groups. He is a Senior Member of CCF and CSIG and an Elected Member of MSA-TC of the IEEE CAS Society. He received the 2009 IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY Best Paper Award and the VCIP 2016 Best 10% Paper Award. He and his students were winners of several technical challenges held in ICCV 2019, ACM MM 2019, ACM MM 2018, ECCV 2018, CVPR 2018, and ICME 2016. He has served as the Registration Co-Chair of ICME 2019 and the Symposium Co-Chair of WCSP 2014.

**Wenhan Yang** (Member, IEEE) received the B.S. and Ph.D. (Hons.) degrees in computer science from Peking University, Beijing, China, in 2012 and 2018, respectively. He was a Visiting Scholar with the National University of Singapore from September 2015 to September 2016 and from September 2018 to November 2018. He is currently a Postdoctoral Research Fellow with the Department of Computer Science, City University of Hong Kong. His current research interests include deep learning-based image processing, bad weather restoration, related applications, and theories.

**Sifeng Xia** (Student Member, IEEE) received the B.S. degree in computer science from Peking University, Beijing, China, in 2017, where he is currently pursuing the master's degree with the Wangxuan Institute of Computer Technology. His current research interests include deep learning-based image processing and video coding.

**Xiaoshuai Zhang** received the B.S. degree in machine intelligence from Peking University, Beijing, China in 2019. He is currently pursuing the Ph.D. degree in computer science and engineering with UC San Diego. His current research interests include 3D Vision, low-level computer vision, and generative models.



**Yuanying Dai** received the B.S. degree in electronic engineering from the Hefei University of Technology in 2016 and the M.S. degree from the University of Science and Technology of China in 2019. Her research interests mainly include video compression and processing.