JPEG Artifacts Removal via Contrastive Representation Learning

Xi Wang, Xueyang Fu*, Yurui Zhu, and Zheng-Jun Zha

University of Science and Technology of China, Hefei, China
{wangxxi, zyr}@mail.ustc.edu.cn, {xyfu, zhazj}@ustc.edu.cn
https://github.com/wang-xi-1/JPEG

Abstract. To meet the needs of practical applications, current deep learning-based methods focus on using a single model to handle JPEG images with different compression qualities, while few of them consider the auxiliary effects of the compression quality information. Recently, several methods estimate quality factors in a supervised learning manner to guide their network to remove JPEG artifacts. However, they may fail to estimate unseen compression types, affecting the subsequent restoration performance. To remedy this issue, we propose an unsupervised compression quality representation learning strategy for the blind JPEG artifacts removal. Specifically, we utilize contrastive learning to obtain discriminative compression quality representations in the latent feature space. Then, to fully exploit the learned representations, we design a compression-guided blind JPEG artifacts removal network, which integrates the discriminative compression quality representations in an information lossless way. In this way, our single network can flexibly handle various JPEG compression images. Experiments demonstrate that our method can adapt to different compression qualities to obtain discriminative representations and outperform state-of-art methods.

Keywords: JPEG Artifacts Removal, Unsupervised Representation Learning, Contrastive Learning, Image Restoration

1 Introduction

Due to the explosive growth of images and videos on the website, lossy compression has become a widely adopted strategy to save transmission bandwidth and storage. JPEG compression [1], which uses discrete cosine transform (DCT), is a popular compression standard due to the ease and speed of its application. First, the JPEG compression divides the image into 8×8 blocks. Then a discrete cosine transform is implemented to obtain DCT coefficients. After the critical lossy step of quantizing and rounding the coefficients on each block, the information is lost, and complex artifacts inevitably appear in the compressed images. These artifacts not only cause visual discomfort but also lead to the performance degradation of subsequent computer vision tasks.

* Corresponding author

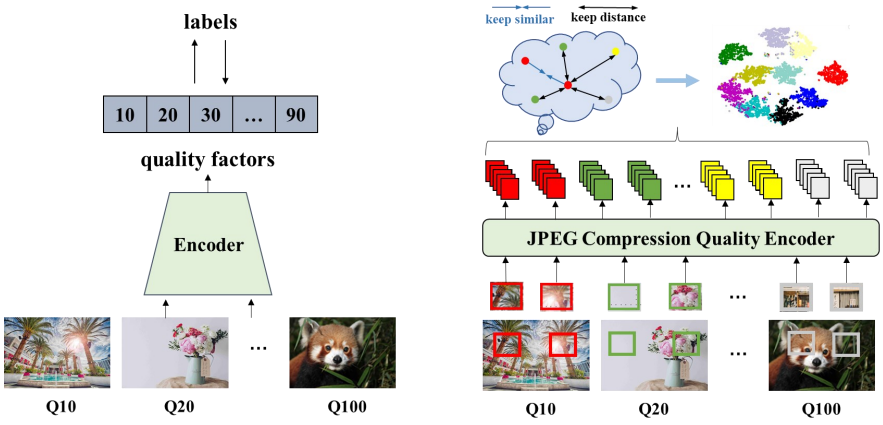


Fig. 1: An illustration of the other **supervised** method with our **unsupervised** compression quality representation learning method. In our method, the same color represents the same compression quality. We use the t-SNE [3] approach to cluster the output compression quality representations.

To mitigate the impact of JPEG compression artifacts, many methods have been proposed. Generally, these methods can be roughly divided into model-based methods and deep learning (DL)-based methods. Model-based methods are primarily based on the filter design [2], they are usually limited to solving certain artifacts (*e.g.*, blocking and ringing artifacts). In recent years, thanks to the rapid development of deep learning network, which has powerful non-linear mapping capabilities, DL-based methods achieve better performance and dominate the field of JPEG artifacts removal.

However, most of the existing DL-based methods [4–7] train a specific network for each compression quality, which significantly limits the practicability of the network. Several blind JPEG compression artifacts removal methods [8, 9] employ a single model to handle different compression qualities. However, these methods ignore the compression quality, and thus cannot explicitly reflect the degradation degree. Parts of the methods take into account compression qualities, but they all have certain shortcomings. For example, DCT-based methods [10, 11] use a quantization table to guide the restoration of the image, but when the image is compressed multiple times, the quantization table information is incomplete. Wang *et al.* [12] utilize the ranker of image compression qualities but treat them to design loss functions instead of adding them into the JPEG artifacts removal network, which cannot fully exploit the distinguishable compression quality information. Jiang *et al.* [13] predicte quality factors in a supervised way which requires label quality factors. But the supervised manner

is difficult to generalize to the unseen compression quality. When the prediction deviates from the accurate compression, the recovery performance will drop.

Unlike previous approaches, we manage to obtain the discriminative compression quality representation rather than predict the exact quality factor. Motivated by the success of contrastive learning [14–16], we propose an unsupervised contrastive learning strategy to obtain compression quality representations, which can fully mine the discrepancy between different compression qualities. Specifically, as shown in Fig. 1, we learn discriminative compression quality representations in the latent feature space by utilizing the variations of different JPEG compression images. In order to fully exploit this information, the learned representations are integrated into the JPEG artifacts removal network in an information lossless way to guide the network training. In this way, our network is able to flexibly process JPEG images with different compression qualities. Compared with directly predicting quality factors in a supervised way, our method does not require ground truth information of specific quality factors, which is accomplished in an unsupervised manner. Therefore, our method has a better generalization ability so that it is more applicable to unseen compression qualities, *e.g.*, real-world scenes(Fig. 2). Not only seen images but also unseen compressed images can be well recovered using our method.

The main contributions of our paper are as follows:

1. We propose a new framework for blind JPEG artifacts removal By taking advantage of the potential compression quality information in JPEG compressed images, our model can work well with all compressed quality JPEG images.

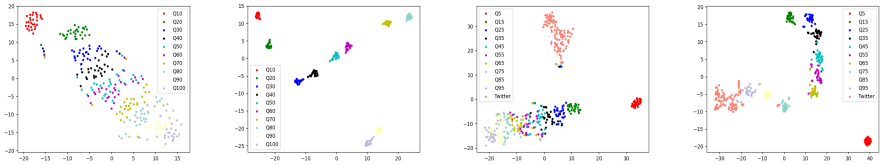
2. We propose an unsupervised manner to extract the discriminative compression quality representation hidden in the JPEG images, then integrate these learned representations into the compression quality-guided JPEG artifacts removal network in an information lossless way to guide the restoration of images with different compression qualities.

3. Experiments demonstrate that our network can flexibly handle various compression qualities and achieve state-of-the-art performance both in seen and unseen JPEG images, *e.g.*, improving 0.3dB in terms of PSNR on the widely used BSDS500 dataset of RGB channels.

2 Related Work

2.1 JPEG Artifacts Removal

There are mainly model-based and DL-based methods for JPEG artifacts removal. The earlier methods perform filtering operations to achieve compression artifacts removal. Foi *et al.* [2], based on shape-adaptive transformations provide image filtering algorithms, clean edges are reconstructed, and no introduce unpleasant ring artifacts. Because it has a natural ill-posed characteristic, Probabilistic-Prior Methods play an important role. Many effective priors, *e.g.*, non-local similarity [17], low-rank [18,19], sparse coding [20], and adaptive DCT transformations [2], are explored. In recent years, DL-based methods have made



(a) E2 layer (b) E3 layer (c) E2 layer (d) E3 layer

Fig. 2: (a)(b) Visualization of different compression quality representations for LIVE1 with quality in **[10, 100]** in steps of 10. (c)(d) **Generalization Capabilities** Visualization of **unseen** compression quality representations for LIVE1 with quality in **[5, 95]** in steps of 10 and the **real-world** dataset(Twitter).

significant progress in JPEG artifacts removal due to the powerful nonlinear mapping capability. ARCNN [4], proposed by Dong *et al.*, is a pioneering work that uses only four layers of CNN. Wang *et al.* [21] introduce a DCT domain prior to facilitating the JPEG artifacts removal. Mao *et al.* [22] use a deep encoding-decoding structure to exploit the rich dependencies of deep features. Some work also embeds traditional priors into deep networks, *e.g.*, multi-scale constraints [5] and wavelet signal structures [7]. Zhang *et al.* [8] achieve blind JPEG artifact removal using BN [23] and residual learning [24]. Since GANs [25] can be used to generate realistic textures, Galteri *et al.* [26] demonstrate that the GAN is able to produce more realistic details than MSE or SSIM based networks. Ehrlish *et al.* [10] also utilize the GAN loss to generate significantly more visually pleasing results. Zhang *et al.* [27] achieve effective image restoration by super-imposing local and non-local attention blocks to construct a residual non-local attention network. Zini *et al.* [9] exploit RRDB to remove JPEG artifacts in a blind way. Recently, there have been some attempts to use compressed quality information. Kim *et al.* [28] utilize the estimated quality factor for JPEG artifacts removal. AGARNet [29] estimates the pixel-wise quality factor in achieving using a single network to cover a wide range of quality factors. Wang *et al.* [12] propose compression quality ranker-guided networks. Jiang *et al.* [13] use a supervised way to predict the compression quality factor directly and embed the predicted quality factor into the subsequent network to guide the JPEG artifacts removal.

Although some methods take the compression qualities into account, they do not fully exploit this information or are limited by the supervised learning method. We propose an unsupervised compression quality representation learning strategy and make adequate exploitation of the learned representations to achieve restoration of all compressed quality JPEG images.

2.2 Contrastive Learning

Unsupervised learning [30–35] is a popular learning technique that does not rely on the label. Unsupervised contrastive learning [36–38] is the most popular method for generating discriminative representation via distinguishing positive

and negative samples in an unsupervised way. In computer vision tasks, there are many flexible choices of positive and negative samples, which allows for the great application of contrastive learning. Although contrastive learning has been widely used in high-level tasks, it has not been widely applied in low-level tasks, especially in the field of JPEG artifacts removal. In this paper, we utilize contrast learning to obtain discriminative compression quality representations to guide our single model in processing JPEG images of all compression qualities.

3 Proposed Method

We propose a blind JPEG artifacts removal network, which consists of two parts: Unsupervised Compression Quality Encoder and Compression Quality-guided JPEG Artifacts Removal Network, as shown in Fig. 3. Our network is trained in two stages. First, we train a compression quality encoder in an unsupervised way to generate discriminative representations for different compression qualities. Second, based on the learned compression quality representations, we design the compression quality-guided JPEG artifacts removal network. In the next section, we describe the network structure and training strategy in detail.

3.1 Unsupervised Compression Quality Encoder

The goal of unsupervised representation learning [35,36] is to learn an encoder that converts input data to general-purpose representations. Unsupervised contrastive learning [30,37,38], which is trained by positive and negative samples, intends to generate similar representations for similar data and to make the representations of different data as different as possible. In order to achieve this goal, the InfoNCE loss [32] is often used to measure the similarity of representations, which uses the dot product measure of similarity:

$$L_q = -\log \frac{\exp(q \cdot k^+/\tau)}{\sum_{i=1}^{N_{neg}} \exp(q \cdot k_i^-/\tau)}, \tag{1}$$

where k^+ denotes a positive sample similar to q, k^- denotes a negative sample not similar to q, $\cdot$ represents the dot product, N_{neg} is the total number of negative samples and τ is a temperature hyper-parameter.

In this paper, we use the unsupervised contrast learning method to extract discriminative compression quality representations of JPEG images. To achieve this goal, we set the patch on the same image to have the same JPEG compression quality and the patch from the different images to have different JPEG compression qualities. Multiple patches can be cropped from each image, where patches from the same image can be used as positive samples, while patches from different images can be used as negative samples.

In the training phase of the compression quality encoder, we randomly select a mini-batch consisting of B images with different compression qualities. Then, two patches are randomly cropped from each JPEG compression image, denoted

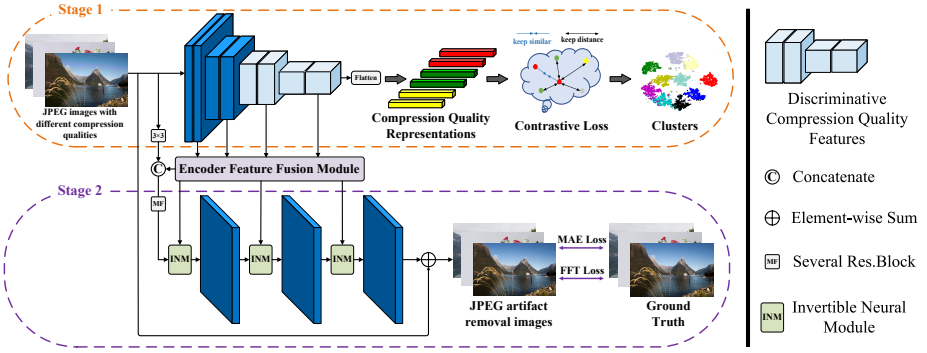


Fig. 3: The architecture of the proposed framework for blind JPEG artifacts removal. The training is divided into two stages. **First**, we train the compression quality encoder and generate discriminative compression quality representations. **Second**, we integrate the learned discriminative compression quality representations into the JPEG artifacts removal network in an information lossless way to handle various JPEG compression images flexibly.

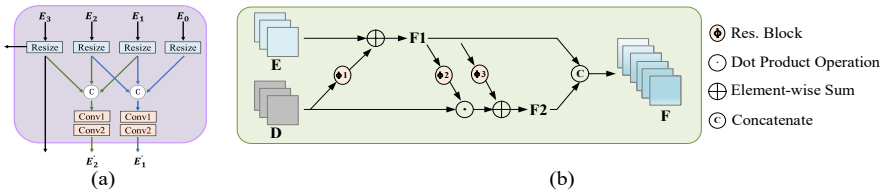


Fig. 4: Multi-scale Information Lossless Fusion Module. It consists of two parts: (a) Encoder Feature Fusion Module, (b) Invertible Neural Module.

as p_i^1 and p_i^2, where p_i indicates that the patch is from the i^{th} JPEG compression image. Then they are fed into the compression quality encoder to get compression quality representations c_i^1 and c_i^2. For each image, we set c_i^1 as a query and c_i^2 as a positive sample, and the compression representation c_j^1 and c_j^2 (i≠j) of patches from other JPEG compressed quality images as negative samples. c_i^1 should be as similar to c_i^2 as possible and as different from c_j^1 and c_j^2 as possible. Recent studies have shown that a large number of negative samples are crucial in unsupervised contrast representation learning. Following MoCo [16], we utilize a queue to store negative samples. The queue stores multiple representations of recent training images, and is dynamically, constantly updating, with representations of the latest images entering the queue and representations of the oldest images leaving the queue. The loss function of the compression quality encoder is:

$$L_{CQE} = \sum_{i=1}^{B} -\log \frac{\exp(c_i^1 \cdot c_i^2/\tau)}{\sum_{j=1}^{N_{neg}} \exp(c_i^1 \cdot c_j^{1,2}/\tau)}, \tag{2}$$

where the numerator represents the dot product of query and positive sample, and the denominator represents the dot product of query and all negative samples in the queue, where the dot product is used to measure the relative distance.

We use a multi-scale feature extraction network as an encoder network, as shown in Fig. 3. The output of each scale of the compression quality encoder is denoted as E_0, E_1, E_2 and E_3, respectively. We provide the detailed network structure in the supplementary material. To demonstrate that our compression quality encoder learns discriminative representations, we visualize the features in the compression quality encoder network using the t-SNE [3] method, as shown in Fig. 2. Our network can obtain discriminative representations on various compression qualities, including unseen degradation types.

3.2 Compression Quality-Guided JPEG Artifact Removal Network

After obtaining compression quality representations, to fully exploit this information, we design a compression quality-guided JPEG artifacts removal network, which integrates the learned compression quality representations in an information lossless way. The network contains multi-scale information lossless feature fusion module and restoration decoder, as shown in Fig. 3.

Multi-scale Information Lossless Fusion Module. In order to better integrate the discriminative representations learned by the compression quality encoder into the subsequent JPEG artifacts removal network, we use a multi-scale information lossless fusion module for this operation. First, since the features at different scales of the network are closely related, we try to fuse feature maps from multi-scales in the comparison quality encoder as much as possible. Specifically, we feed the output of each scale into the Encoder Feature Fusion Module (EFFM) and resize them to the same scales. E_1 and E_2 incorporate feature maps from nearby scales. If these feature maps were to be directly concatenated into the network afterward, this would result in a large number of operations, so to reduce the computational effort we introduce two convolution operations to fuse them, the output feature maps are denoted $E_1^{'}$ and $E_2^{'}$. The formulations are as:

$$E_1^{'} = EFFM[E_0, E_1, E_2], \tag{3}$$

$$E_2^{'} = EFFM[E_1, E_2, E_3], \tag{4}$$

where $EFFM$ includes convolution and resizes operations, $[\cdot]$ represents concatenation along the channel dimension, as shown in Fig. 4(a).

In order to fully exploit the learned discriminative compression quality feature representations, we use invertible fusion modules designed based on invertible neural architecture [39, 40] to preserve all information about the input features. Compared to simple concatenation operations, the invertible neural network [41–43] is information lossless in the processes of the transformation. In our work, a total of three invertible fusion modules are used, corresponding to

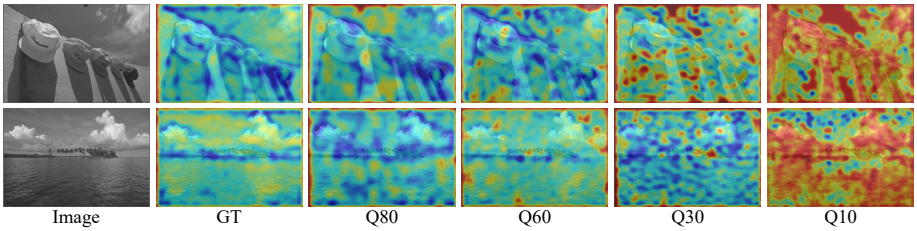


Fig. 5: Visualization of the intermediate feature maps of our Compression Quality Encoder at JPEG images with different compression qualities.

the outputs of the last three scales in the compression quality encoder. Invertible networks require the input to be divided into two parts, we set the feature maps from the EFFM and subsequent restoration decoder as inputs, noted as E and D, respectively. Take one module as an example, it performs the following operations:

$$F_1 = E + \phi_1(D), \tag{5}$$

$$F_2 = D \odot \exp(\phi_2(F_1)) + \phi_3(F_1), \tag{6}$$

$$F = Concat(F_1, F_2), \tag{7}$$

where $\exp(\cdot)$ and $\odot$ indicate exponential function and dot product operation, respectively. As shown in Fig. 4(b), we choose residual blocks to perform ϕ_1, ϕ_2 and ϕ_3, each residual block is composed of two 3×3 convolutions layers with the LeakyReLU activation function [44] in the middle.

Restoration Decoder. The outputs of each invertible neural module are fed into the local recovery module referenced from the RNAN [27]. The network then applies a 1×1 convolution layer to restore the feature maps to the original image channel. Finally, we use global residual learning to connect the input and output images to achieve faster training.

3.3 Loss function

MAE Loss. We adopt the Mean Absolute Error (MAE) loss to reduce the distance between the predicted image I_{pre} and the ground truth I_{gt}, which is defined as:

$$L_{MAE} = \frac{1}{N} \sum_{i=1}^{N} \parallel I_{pre}^{i} - I_{gt}^{i} \parallel_1, \tag{8}$$

where N is the number of training samples within a mini-batch.

FFT loss. Since the quantization operation of JPEG compression results in the loss of high-frequency information in the image, we further employ the difference

Table 1: Quantitative comparisons of different methods on **grayscale** JPEG images. PSNR / SSIM / PSNR-B format. The best and the second best results are **boldfaced** and underlined, respectively.

Dataset	Quality	JPEG	ARCNN [4]	DnCNN [8]	MWCNN [7]	DCSC [46]
Classic5	10	27.82/0.760/25.21	29.03/0.793/28.76	29.40/0.803/29.13	30.01/0.820/29.59	29.62/0.810/29.30
	20	30.12/0.834/27.50	31.15/0.852/30.59	31.63/0.861/31.19	32.16/0.870/31.52	31.81/0.864/31.34
	30	31.48/0.867/28.94	32.51/0.881/31.98	32.91/0.886/32.38	33.43/0.893/32.62	33.06/0.888/32.49
	40	32.43/0.885/29.92	33.32/0.895/32.79	33.77/0.900/33.23	34.27/0.906/33.35	33.87/0.902/33.30
LIVE1	10	27.77/0.773/25.33	28.96/0.808/28.68	29.19/0.812/28.90	29.69/0.825/29.32	29.34/0.818/29.01
	20	30.07/0.851/27.57	31.29/0.873/30.76	31.59/0.880/31.07	32.04/0.889/31.51	31.70/0.883/31.18
	30	31.41/0.885/28.92	32.67/0.904/32.14	32.98/0.909/32.34	33.45/0.915/32.80	33.07/0.911/32.43
	40	32.35/0.904/29.96	33.61/0.920/33.11	33.96/0.925/33.28	34.45/0.930/33.78	34.02/0.926/33.36
BSDS500	10	27.80/0.768/25.10	29.10/0.804/28.73	29.21/0.809/28.80	29.61/0.820/29.14	29.32/0.813/28.91
	20	30.05/0.849/27.22	31.28/0.870/30.55	31.53/0.878/30.79	31.92/0.885/31.15	31.63/0.880/30.92
	30	31.37/0.884/28.53	32.67/0.902/31.94	32.90/0.907/31.97	33.30/0.912/32.34	32.99/0.908/32.08
	40	32.30/0.903/29.49	33.55/0.918/32.78	33.85/0.923/32.80	34.27/0.928/33.19	33.92/0.924/32.92

Dataset	Quality	RNAN [27]	RDN [47]	QGAC [48]	FBCNN [13]	Ours
Classic5	10	29.96/0.819/29.42	30.03/0.819/29.59	29.84/0.812/29.43	30.12/**0.822**/29.80	**30.16/0.822/29.85**
	20	32.11/0.869/31.26	32.19/0.870/31.53	31.98/0.869/31.37	32.31/0.872/31.74	**32.37/0.873/31.84**
	30	33.38/0.892/32.35	33.46/0.893/32.59	33.22/0.892/32.42	33.54/0.894/32.78	**33.60/0.895/32.89**
	40	34.27/0.906/33.40	-	34.05/0.905/33.12	34.35/0.907/33.48	**34.43/0.908/33.58**
LIVE1	10	29.63/0.824/29.13	29.70/0.825/29.37	29.51/0.825/29.13	29.75/0.827/29.40	**29.80/0.827/29.44**
	20	32.03/0.888/31.12	32.10/0.889/31.29	31.83/0.888/31.25	32.13/0.889/31.57	**32.19/0.890/31.63**
	30	33.45/0.915/32.22	33.54/0.916/32.62	33.20/0.914/32.47	33.54/0.916/32.83	**33.62/0.918/32.91**
	40	34.47/0.930/33.66	-	34.16/0.929/33.36	34.53/0.931/33.74	**34.62/0.931/33.84**
BSDS500	10	29.08/0.805/28.48	29.24/0.808/28.71	29.46/0.821/28.97	29.67/0.821/29.22	**29.70/0.822/29.27**
	20	31.25/0.875/30.27	31.48/0.879/30.45	31.73/0.884/30.93	32.00/0.885/31.19	**32.06/0.886/31.27**
	30	32.70/0.907/31.33	32.83/0.908/31.60	33.07/0.912/32.04	33.37/0.913/32.32	**33.45/0.914/32.41**
	40	33.47/0.923/32.27	-	34.01/0.927/32.81	34.33/0.928/33.10	**34.42/0.929/33.22**

between the predicted image and the ground truth in the frequency domain [45] to optimize our network. The frequency loss is defined as:

$$L_{\mathrm{FFT}} = \frac{1}{N}\sum_{i=1}^{N} \|\mathrm{FFT}(I^i_{pre}) - \mathrm{FFT}(I^i_{gt})\|_1, \tag{9}$$

where FFT stands for fast Fourier transform, which converts an image to the frequency domain. The total loss function is defined as:

$$L_{total} = L_{MAE} + \lambda L_{\mathrm{FFT}}. \tag{10}$$

In our experiment, we set λ equal to 0.1.

4 Experiments

4.1 Experimental Datasets and Implementation details

Datasets. In our experiments, we use a total of six datasets: DIV2K [49], BSDS500 [50], LIVE1 [51], Classic5 [52], ICB [48] and Twitter [4]. 900 images

Table 2: Perceptual metrics results of LIPIS↓ / FID↓.

Dataset	Classic5				LIVE1			
Quality	Q10	Q20	Q30	Q40	Q10	Q20	Q30	Q40
FBCNN	**0.1543**/103.85	0.1072/46.98	0.0817/31.53	0.0665/23.66	**0.1603**/69.47	0.0924/32.78	0.0637/21.87	0.0479/15.50
Ours	0.1545/**101.68**	**0.1062/43.08**	**0.0806/30.19**	**0.0651/22.44**	0.1633/**65.92**	**0.0923/32.03**	**0.0630/20.79**	**0.0469/14.72**

from the training and validation sets of DIV2K and 200 images from the training sets of BSDS500 are used for training. The test set of BSDS500, Classic5, LIVE1, ICB and Twitter are used for testing. We used the Y channel of YCbCr space for grayscale image recovery and the RGB channel for color image recovery.

Training Settings. The compression quality of the training images is set to [Q10, Q90] at step 10 and we randomly crop 256×256 patches from the images. Note that our model is trained in two stages. For the first stage, when we train the compression quality encoder, the learning rate is set to 0.001, and the number of training epochs is set to 200, then we freeze the model weights. For the second stage of training the JPEG artifacts removal network, the initial learning rate is set to 0.0001 and decayed by a cosine annealing algorithm with T = 600. For the optimization model, we set the epochs for 600 with a batch size of 8 and choose the Adam optimizer [53] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. In addition, our single model can handle multiple JPEG compression qualities. We train our model on two NVIDIA GeForce GTX 3090 GPUs by using PyTorch.

Testing Settings. For grayscale images, we evaluate the performance of our model on Classic5 [52], LIVE1 [51], Twitter [4] and the test set of BSDS500 [50]. During the standard testing phase, all test datasets are all applied JPEG compression with compression quality factors of Q10, Q20, Q30 and Q40. During the testing phase of the model generalizability capability, these test datasets are compressed into Q15, Q25, Q35 and Q45. For color images, we do not use the Classic5 [52] but the ICB [48] instead.

Evaluation Metrics. We use PSNR, SSIM(structural similarity) [54], and PSNR-B(specially designed for JPEG artifacts removal) [55] to quantitatively assess the performance of our JPEG artifacts removal model.

4.2 Experiments on Synthetic Datasets

Feature Maps Visualisation for Compression Quality Encoder. To demonstrate the ability of our compression quality encoder to distinguish different quality factors, we perform Grad-CAM [56] to visualize the learned feature maps in Fig. 5. It is clear that our compression quality encoder generates different feature maps for different compression qualities, which can provide discriminative information to guide subsequent JPEG artifacts removal.

Table 3: Quantitative comparisons of different methods on **color** JPEG images. PSNR / SSIM / PSNR-B format. The best and the second best results are **boldfaced** and underlined, respectively.

Dataset	Quality	JPEG	QGAC [48]	FBCNN [13]	Ours
*LIVE*1	10	25.69/0.743/24.20	27.62/0.804/27.43	27.77/0.803/27.51	**27.80/0.805/27.57**
	20	28.06/0.826/26.49	29.88/0.868/29.56	30.11/0.868/29.70	**30.23/0.872/29.85**
	30	29.37/0.861/27.84	31.17/0.896/30.77	31.43/0.897/30.92	**31.58/0.900/31.13**
	40	30.28/0.882/28.84	32.05/0.912/31.61	32.34/0.913/31.80	**32.53/0.916/32.04**
*BSDS*500	10	25.84/0.741/24.13	27.74/0.802/27.47	27.85/0.799/27.52	**27.91/0.803/27.59**
	20	28.21/0.827/26.37	30.01/0.869/29.53	30.14/0.867/29.56	**30.31/0.872/29.74**
	30	29.57/0.865/27.72	31.33/0.898/30.70	31.45/0.897/30.72	**31.69/0.901/30.96**
	40	30.52/0.887/28.69	32.25/0.915/31.50	32.36/0.913/31.52	**32.66/0.918/31.82**
ICB	10	29.44/0.757/28.53	32.06/**0.816**/32.04	**32.18**/0.815/**32.15**	32.05/0.813/32.04
	20	32.01/0.806/31.11	34.13/0.843/34.10	**34.38/0.844/34.34**	34.32/0.842/34.31
	30	33.20/0.831/32.35	35.07/**0.857**/35.02	**35.41/0.857/35.35**	35.37/0.856/**35.35**
	40	33.95/0.840/33.14	32.25/**0.915**/31.50	**36.02**/0.866/35.95	35.99/0.860/**35.97**

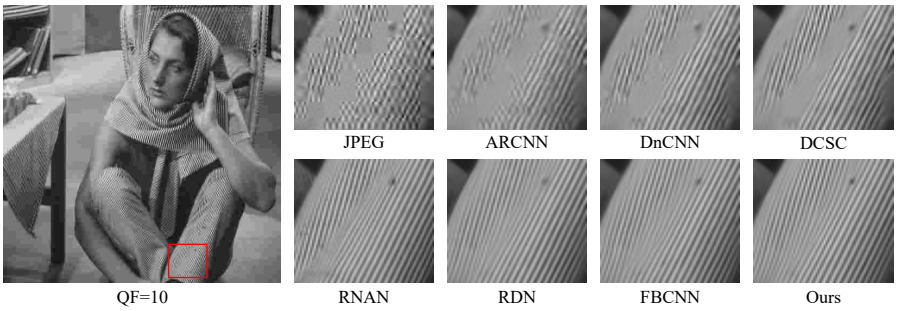


Fig. 6: Visual comparisons of JPEG image "Classic5: barbara" with QF=10.

Y Channel JPEG Artifacts Removal. We first evaluate the effect of our model on the Y-channel JPEG compressed images. For LIVE1 [51], Classic5 [52], BSDS500 [50], we compared our model with a series of JPEG artifact removal network: *i.e.*, ARCNN [4], DnCNN [8], MWCNN [7], DCSC [46], RNAN [27], RDN [47], QGAC [48] and FBCNN [13]. For quantitative evaluation, we use PSNR, SSIM and PSNR-B, the results of them are presented in Table 1. As can be seen that our proposed model outperforms all previous methods. This proves the validity of our proposed model. Note that we use a single model for all compression qualities, this allows for greater flexibility in our models, and our method outperforms all those methods that train one model for one compression quality. We show some visual results of the Classic5 recovery image in Fig. 6, demonstrating the more pleasing visual effect of our method. Moreover, we utilize LIPIS [57] and FID [58] to evaluate the perceptual performance in Table 2.

Table 4: Quantitative comparisons of **Generalization Capabilities**. PSNR / SSIM / PSNR-B format. The best results are **boldfaced**. Our model has not seen the compression quality of the test phase during the training phase.

Dataset	Training Quality(step)	Testing Quality	JPEG	FBCNN [13]	Ours
*Classic*5	$Q10 - Q90(10)$	15	29.17/0.807/26.53	**31.42**/**0.854**/**30.97**	31.37/**0.854**/30.86
		25	30.87/0.853/28.30	33.02/**0.885**/32.34	**33.04**/**0.885**/**32.41**
		35	32.01/0.877/29.50	33.99/0.9015/33.20	**34.05**/**0.902**/**33.30**
		45	32.84/0.892/30.37	34.72/0.9122/33.83	**34.79**/**0.913**/**33.92**
*LIVE*1	$Q10 - Q90(10)$	15	29.13/0.822/26.65	**31.15**/0.866/**30.69**	31.12/**0.867**/30.53
		25	30.81/0.871/28.29	32.91/**0.905**/**32.26**	**32.95**/**0.905**/**32.26**
		35	31.93/0.896/29.48	34.09/**0.925**/33.33	**34.15**/**0.925**/**33.40**
		45	32.778/0.912/30.43	34.96/0.936/34.12	**35.04**/**0.937**/**34.22**
*BSDS*500	$Q10 - Q90(10)$	15	29.13/0.819/26.34	**31.04**/**0.862**/**30.39**	30.99/**0.862**/30.22
		25	30.77/0.869/27.93	32.75/0.901/31.81	**32.79**/**0.902**/**31.83**
		35	31.88/0.895/29.05	33.91/**0.922**/32.76	**33.97**/**0.922**/**32.85**
		45	32.73/0.911/29.94	34.76/0.934/33.46	**34.85**/**0.935**/**33.58**

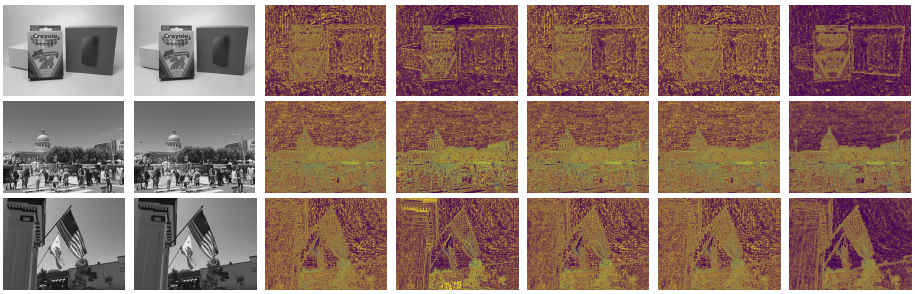


Fig. 7: Visual comparisons on **real-world** images from "Twitter" dataset.

RGB Channels JPEG Artifacts Removal. To evaluate the effectiveness of our model on color images, we also trained our model on color images. We set the number of input and output channels to 3, while the other model settings remain unchanged. The test data set is selected LIVE1 [51] and test sets of BSDS500 [50]. Quantitative results are shown in Table 3. It can be seen that our method achieves better JPEG artifacts removal results on color images as well.

Study of Generalization Capabilities. Both our compression quality encoder and JPEG artifacts removal network are trained only on training data with compression quality set to [Q10, Q90] at step 10. To explore whether our model can perform well on unseen JPEG compressed quality images, we choose images with compression qualities of Q15, Q25, Q35 and Q45. As shown in Table 4, our single model consistently performs well on unseen compression qualities. All these processes are performed on the Y-channel of LIVE1 [51], Classic5 [52] and the test set of BSDS500 [50].

Table 5: Quantitative **Ablation Analysis** on PSNR/SSIM/PSNR-B Values.The dataset used in this experiment is Classic5.

Models	CQE	CQE (pre-trained)	MILFM		FFT Loss	Compression Quality			
			INM	EFFM		Q10	Q20	Q30	Q40
model-1					✓	29.97/0.818/29.63	32.21/0.871/31.68	33.48/0.893/32.76	34.31/0.906/33.46
model-2	✓		✓	✓	✓	30.04/0.820/29.73	32.26/0.871/31.78	33.52/0.894/32.85	34.36/0.907/33.57
model-3(a)	✓	✓		✓	✓	30.11/0.821/29.77	32.30/0.872/31.74	33.55/0.894/32.81	34.39/0.907/33.52
model-3(b)	✓	✓			✓	30.09/0.821/29.72	32.30/0.872/31.72	33.54/0.894/32.78	34.37/0.907/33.48
model-4	✓	✓	✓	✓		30.03/0.819/29.66	32.21/0.871/31.69	33.50/0.893/32.77	34.32/0.906/33.51
Ours	✓	✓	✓	✓	✓	30.16/0.822/29.85	32.37/0.873/31.84	33.60/0.895/32.89	34.43/0.908/33.58

4.3 Experiments on Real-World Compression Qualities

To avoid taking up too much storage and transmission resources, social platforms such as Twitter often compress uploaded images, which inevitably reduces visual feelings of users. To test the performance of our model on real data, we use the Twitter dataset to test the real image directly using the model we trained on the synthetic datasets. Since the real images were too large in resolution, we first crop images and then feed them into the network. We show in Fig. 7 the visual residual maps of other methods and ours to increase the distinction of the visualization. Note that the residual map means the difference between the estimated result and its ground truth. It is clear that our method achieves the better visual result. This result shows that our method works better than other methods on unseen JPEG compressed quality images.

4.4 Ablation Analysis

We remove some parts of the network that we designed and report their effect. We choose Q10 of the Classic5 dataset to report the results. For all ablation experiments, quantitative results are presented in Table 5.

Effect of Compression Quality Encoder(CQE). The compression quality encoder generates discriminative representations that provide recovery guidance for subsequent JPEG artifacts removal networks. To demonstrate the effectiveness of compression quality encoder, we compare two-stage joint training and two-stage separate training strategies: (1) remove the entire compression quality encoder, denoted as model-1, (2) train the entire network directly without removing the compression quality encoder, but without pre-training, denoted as model-2. The results of the quantitative evaluation show that networks that remove the compression quality representation learning encoder would cause the network performance to drop. Moreover, we show the second option in which the compression quality encoder visualizes the clustered feature maps with and without the pre-trained weights in the supplementary material.

Effect of Multi-scale Information Lossless Fusion Module(MILFM). With the compression quality encoder and the extracted compression quality

feature representations, we utilize the multi-scale information lossless feature fusion module to integrate them with the subsequent JPEG artifacts removal network. To demonstrate the effectiveness of this fusion module, we replace INM with the concatenation operation and convolution layers, denoted as model-3(a). Moreover, we replace INM and EFFM with the concatenation operation and convolution layers to achieve feature fusion, denoted as model-3(b). In this way, the network does not make much difference in terms of the number of parameters. As can be seen from the PSNR values taken, the performance of the model will drop if the concatenation operation and convolution layers are used as the fusion module. On the contrary, better JPEG artifacts removal results can be achieved by using the multiscale information lossless fusion module.

Effect of FFT loss. In order to better recover the lost high-frequency information, we introduced the FFT Loss. To test the capability of this loss function, we removed this Loss without changing the other parts of the model, which was noted as model-4. It is seen from the experimental results that the recovery of the model decreased due to the disappearance of the FFT loss.

5 Conclusions

In this paper, we propose an unsupervised JPEG compression quality representation learning to guide the blind JPEG artifacts removal. Rather than directly predicting the exact quality factor, our approach focuses on mining the discrepancy in compression quality of various compressed images. Moreover, to fully exploit the learned representations, we design a compression-guided blind JPEG artifacts removal network, which specially integrates the learned discriminative compression quality representations in an information lossless way. Experiments demonstrate that our unsupervised compression quality learning strategy could extract discriminative representations, and our network achieves state-of-the-art performances for various types of JPEG compressed quality images.

Acknowledgement This work was supported by the National Key R&D Program of China under Grant 2020AAA0105702, the National Natural Science Foundation of China (NSFC) under Grants U19B2038 and 61901433, the University Synergy Innovation Program of Anhui Province under Grants GXXT-2019-025, the Fundamental Research Funds for the Central Universities under Grant WK2100000024, and the USTC Research Funds of the Double First-Class Initiative under Grant YD2100002003.

References

1. G. K. Wallace, "The jpeg still picture compression standard," *IEEE transactions on consumer electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992. 1
2. A. Foi, V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive dct for high-quality denoising and deblocking of grayscale and color images," *IEEE transactions on image processing*, vol. 16, no. 5, pp. 1395–1411, 2007. 2, 3
3. L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008. 2, 7
4. C. Dong, Y. Deng, C. C. Loy, and X. Tang, "Compression artifacts reduction by a deep convolutional network," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 576–584. 2, 4, 9, 10, 11
5. L. Cavigelli, P. Hager, and L. Benini, "Cas-cnn: A deep convolutional neural network for image compression artifact suppression," in *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2017, pp. 752–759. 2, 4
6. Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1256–1272, 2016. 2
7. P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo, "Multi-level wavelet-cnn for image restoration," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 773–782. 2, 4, 9, 11
8. K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE transactions on image processing*, vol. 26, no. 7, pp. 3142–3155, 2017. 2, 4, 9, 11
9. S. Zini, S. Bianco, and R. Schettini, "Deep residual autoencoder for blind universal jpeg restoration," *IEEE Access*, vol. 8, pp. 63 283–63 294, 2020. 2, 4
10. M. Ehrlich, L. Davis, S.-N. Lim, and A. Shrivastava, "Quantization guided jpeg artifact correction," in *European Conference on Computer Vision*. Springer, 2020, pp. 293–309. 2, 4
11. J. Guo and H. Chao, "Building dual-domain representations for compression artifacts reduction," in *European Conference on Computer Vision*. Springer, 2016, pp. 628–644. 2
12. M. Wang, X. Fu, Z. Sun, and Z.-J. Zha, "Jpeg artifacts removal via compression quality ranker-guided networks," in *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, 2021, pp. 566–572. 2, 4
13. J. Jiang, K. Zhang, and R. Timofte, "Towards flexible blind jpeg artifacts removal," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 4997–5006. 2, 4, 9, 11, 12
14. T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607. 3
15. A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with convolutional neural networks," *Advances in neural information processing systems*, vol. 27, 2014. 3
16. K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738. 3, 6
17. X. Zhang, R. Xiong, S. Ma, and W. Gao, "Reducing blocking artifacts in compressed images via transform-domain non-local coefficients estimation," in *2012*

IEEE International Conference on Multimedia and Expo. IEEE, 2012, pp. 836–841. 3
18. X. Zhang, R. Xiong, X. Fan, S. Ma, and W. Gao, "Compression artifact reduction by overlapped-block transform coefficient estimation with block similarity," *IEEE transactions on image processing*, vol. 22, no. 12, pp. 4613–4626, 2013. 3
19. J. Ren, J. Liu, M. Li, W. Bai, and Z. Guo, "Image blocking artifacts reduction via patch clustering and low-rank minimization," in *2013 Data Compression Conference*. IEEE, 2013, pp. 516–516. 3
20. H. Chang, M. K. Ng, and T. Zeng, "Reducing artifacts in jpeg decompression via a learned dictionary," *IEEE transactions on signal processing*, vol. 62, no. 3, pp. 718–728, 2013. 3
21. 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2764–2772. 4
22. X. J. Mao, C. Shen, and Y. B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," *IEEE transactions on image processing.*, vol. 15, no. 13, pp. 3142–3155, 2017. 4
23. S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456. 4
24. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778. 4
25. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014. 4
26. L. Galteri, L. Seidenari, M. Bertini, and A. Del Bimbo, "Deep generative adversarial compression artifact removal," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4826–4835. 4
27. Y. Zhang, K. Li, K. Li, B. Zhong, and Y. Fu, "Residual non-local attention networks for image restoration," *arXiv preprint arXiv:1903.10082*, 2019. 4, 8, 9, 11
28. Y. Kim, J. W. Soh, J. Park, B. Ahn, H.-S. Lee, Y.-S. Moon, and N. I. Cho, "A pseudo-blind convolutional neural network for the reduction of compression artifacts," *IEEE Transactions on circuits and systems for video technology*, vol. 30, no. 4, pp. 1121–1135, 2019. 4
29. Y. Kim, J. W. Soh, and N. I. Cho, "Agarnet: adaptively gated jpeg compression artifacts removal network for a wide range quality factor," *IEEE Access*, vol. 8, pp. 20 160–20 170, 2020. 4
30. Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3733–3742. 4, 5
31. C. Zhuang, A. L. Zhai, and D. Yamins, "Local aggregation for unsupervised learning of visual embeddings," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6002–6012. 4
32. A. Van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv e-prints*, pp. arXiv–1807, 2018. 4, 5
33. R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," *arXiv preprint arXiv:1808.06670*, 2018. 4

34. O. Henaff, "Data-efficient image recognition with contrastive predictive coding," in *International Conference on Machine Learning*. PMLR, 2020, pp. 4182–4192. 4
35. P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," *Advances in neural information processing systems*, vol. 32, 2019. 4, 5
36. T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607. 4, 5
37. X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," *arXiv preprint arXiv:2003.04297*, 2020. 4, 5
38. R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," *arXiv preprint arXiv:1808.06670*, 2018. 4, 5
39. L. Dinh, D. Krueger, and Y. Bengio, "Nice: Non-linear independent components estimation," *arXiv preprint arXiv:1410.8516*, 2014. 7
40. L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real nvp," *arXiv preprint arXiv:1605.08803*, 2016. 7
41. Y. Liu, Z. Qin, S. Anwar, P. Ji, D. Kim, S. Caldwell, and T. Gedeon, "Invertible denoising network: A light solution for real noise removal," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 365–13 374. 7
42. S. Zhang, C. Zhang, N. Kang, and Z. Li, "ivpf: Numerical invertible volume preserving flow for efficient lossless compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 620–629. 7
43. Y. Xing, Z. Qian, and Q. Chen, "Invertible image signal processing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6287–6296. 7
44. B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015. 8
45. S.-J. Cho, S.-W. Ji, J.-P. Hong, S.-W. Jung, and S.-J. Ko, "Rethinking coarse-to-fine approach in single image deblurring," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 4641–4650. 9
46. X. Fu, Z. J. Zha, F. Wu, X. Ding, and J. Paisley, "Jpeg artifacts reduction via deep convolutional sparse coding," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 9, 11
47. Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image restoration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 7, pp. 2480–2495, 2020. 9, 11
48. M. Ehrlich, L. Davis, S.-N. Lim, and A. Shrivastava, "Quantization guided jpeg artifact correction," in *European Conference on Computer Vision*. Springer, 2020, pp. 293–309. 9, 10, 11
49. E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 126–135. 9
50. P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 5, pp. p.898–916, 2011. 9, 10, 11, 12
51. H. Sheikh, "Live image quality assessment database release 2," *http://live.ece.utexas.edu/research/quality*, 2005. 9, 10, 11, 12

52. R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *International conference on curves and surfaces*. Springer, 2010, pp. 711–730. 9, 10, 11, 12
53. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. 10
54. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004. 10
55. T. Tadala and S. E. V. Narayana, "A novel psnr-b approach for evaluating the quality of de-blocked images," 2012. 10
56. R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626. 10
57. R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595. 11
58. M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017. 11