JPEG Artifacts Removal via Compression Quality Ranker-Guided Networks

Menglu Wang[1*], **Xueyang Fu**[1*], **Zepei Sun**[2] and **Zheng-Jun Zha**[1†]
[1]University of Science and Technology of China, China
[2]Xidian University, China
vault@mail.ustc.edu.cn, xyfu@ustc.edu.cn, zpsun@stu.xidian.edu.cn, zhazj@ustc.edu.cn

Abstract

Existing deep learning-based image de-blocking methods use only pixel-level loss functions to guide network training. The JPEG compression factor, which reflects the degradation degree, has not been fully utilized. However, due to the non-differentiability, the compression factor cannot be directly utilized to train deep networks. To solve this problem, we propose compression quality ranker-guided networks for this specific JPEG artifacts removal. We first design a quality ranker to measure the compression degree, which is highly correlated with the JPEG quality. Based on this differentiable ranker, we then propose one quality-related loss and one feature matching loss to guide de-blocking and perceptual quality optimization. In addition, we utilize dilated convolutions to extract multi-scale features, which enables our single model to handle multiple compression quality factors. Our method can implicitly use the information contained in the compression factors to produce better results. Experiments demonstrate that our model can achieve comparable or even better performance in both quantitative and qualitative measurements.

1 Introduction

Lossy image compression methods, e.g. JPEG and WebP, have been widely used in Internet information transmission to save bandwidth, reduce storage requirements, and increase upload and download speeds. However, due to the information loss, complex artifacts are inevitably introduced into compressed images, such as blocking artifacts and ringing effects [Dong *et al.*, 2015]. These artifacts not only decrease the visual quality, but also affect the performance of subsequent computer vision systems. Therefore, exploring methods to effectively reduce compression artifacts is urgently needed. Among various compression methods, JPEG compression based on the discrete cosine transform (DCT) is the most commonly used algorithm. The image firstly is divided into blocks with size of 8×8, and then DCT is applied on each block. Finally, the DCT coefficients of each block are quantized and rounded. Some high-frequency components that contain low energy are discarded to reduce the bit rate. Since JPEG compression applies DCT on each block, the correlation between adjacent blocks is ignored, which introduces blocking artifacts. Meanwhile, blurring and ringing effects occur due to the quantization and rounding operations.

To cope with JPEG compression artifacts, many methods have been proposed. In general, the existing methods can be divided into model-based methods and learning-based methods. Previous model-based methods are mostly based on filter design and can only solve certain types of artifacts. For example, Pointwise SA-DCT [Foi *et al.*, 2007] removes blocking artifacts at the cost of introducing blurring effects. In recent years, due to the powerful representation ability, deep learning-based methods have been explored to directly learn a nonlinear mapping from JPEG images to its clean version. For example, AR-CNN [Dong *et al.*, 2015] proposes a relatively shallow network to learn the mapping function. Inspired by deep residual learning [He *et al.*, 2016], DnCNN [Zhang *et al.*, 2017] proposes very deep networks for general image restorations.

Despite the impressive progress of deep learning, there are still some problems in this area. Firstly, most learning-based methods use only pixel-level loss functions, e.g., mean squared error (MSE), to optimize the model. The JPEG compression factor, which reflects the degree of degradation, is ignored. Secondly, existing methods directly use pre-trained VGG [Johnson *et al.*, 2016] or construct adversarial learning [Goodfellow *et al.*, 2014] to improve perceptual quality. However, these strategies are not designed for the specific JPEG artifacts reduction, which leads to a poor trade-off between distortion and perception. Third, most deep learning methods require different models to solve different compression qualities, which limits their practical values.

To address the above problems, we propose compression quality ranker-guided networks for JPEG artifacts reduction. The motivation of our method is that the JPEG compression factor implies the compressed image quality, which should be fully utilized for de-blocking. To address the non-differentiability of directly using compression quality factor, we build a deep ranker to reflect different JPEG compression qualities. Based on this differentiable ranker, we design one

[*] Co-first authors contributed equally
[†] Corresponding author

quality-related loss and one feature matching loss to guide the optimization of both de-blocking and perceptual quality. By combining the above two loss functions with mean absolute error, the training process can be guided from both the semantic level and pixel level. To enable our single model to handle multiple JPEG qualities, we utilize the dilated convolutions to extract multi-scale local features. These task-specific designs allow our method to generate better de-blocking results.

In summary, the contributions of this paper are four-fold:

- We propose a general framework for JPEG artifacts removal by taking compression factors into consideration. The different compression degrees, which are ignored in existing methods, are fully utilized to help design algorithms for this specific de-blocking problem.
- We design a deep ranker to address the non-differentiability of compression factors. Based on this ranker, we propose two loss functions to guide the de-blocking network training from both pixel level and sematic level, which leads to better de-blocking results.
- We incorporate the dilated convolutions into the residual attention block [Zhang *et al.*, 2019b]. In this way, multi-scale local spatial features can be fully explored to generate rich representations, which makes it possible to handle multiple JPEG qualities using a single model.
- Experiments demonstrate that our proposed model reaches comparable or superior performance compared with state-of-the-art approaches both in qualitative and quantitative measurements.

2 Related Work

JPEG artifacts removal. Traditional JPEG artifacts removal methods pay attention to filter design, e.g., Pointwise SA-DCT [Foi *et al.*, 2007] proposes the shape-adaptive DCT-based filter for image denoising and de-blocking. The method [Chang *et al.*, 2013] utilizes sparse coding to restore compressed images. Others treat JPEG artifacts removal as an ill-posed inverse problem and solve it by using regression trees [Jancsary *et al.*, 2012] and non-local self-similarity property [Li *et al.*, 2017]. Due to the powerful representation ability, deep convolutional neural networks (CNNs) are utilized and explored to solve image regression problems [Liu *et al.*, 2018]. For example, the first deep CNNs-based method [Dong *et al.*, 2015] designs a 4-layer fully connected CNNs to learn the mapping function. DeepSD [Vandal *et al.*, 2018] obtains local details information by stacking CNN structures. HRFR [Zhang *et al.*, 2018] restores images at the feature level. Inspired by the excellent performance of deep residual learning [He *et al.*, 2016], very deep network architectures are introduced for image restoration [Zhang *et al.*, 2017]. Inspired by the notable progress of generative adversarial networks (GANs) [Goodfellow *et al.*, 2014], some methods [Guo and Chao, 2017; Zhu *et al.*, 2018; Wang *et al.*, 2019] are proposed to improve the visual quality by photo-realistic details generation. D^3 [Wang *et al.*, 2016] builds a cascaded network in dual-domain to obtain speed and de-blocking performance gains. In [Chen and Pock, 2017], the authors propose a trainable nonlinear reaction diffusion model for image restoration. Recently, the method [Fan *et al.*, 2018] proposes a decoupled learning framework to combine different parameterized operators. DCSC [Fu *et al.*, 2019] introduces a more compact and explainable deep sparse coding architecture to generate high-quality de-blocking results. RNAN [Zhang *et al.*, 2019b] designs residual non-local attention networks for general image restoration, such as image super-resolution and JPEG artifacts removal.

Ranker for image processing. Image quality ranker has proven useful for image processing. RankCGAN [Saquil *et al.*, 2018] designs a Siamese-network-based ranker to handle continuous attribute values for the task of image generation with semantic attributes. RankSRGAN [Zhang *et al.*, 2019a] proposes a rank approach to generate the relative natural image quality evaluator (NIQE) value for image super-resolution. However, these methods design image quality ranker to generate subjective visually pleasing results. While our method aims to design a differentiable ranker related to various JPEG compression factors. This ranker, which fits this specific JPEG artifacts removal problem, can better guide the subsequent de-blocking network training.

3 Methodology

Our proposed framework consists of two components: compression quality ranker (CQR) and de-blocking network, whose architectures are depicted in Figures 1 and 2, respectively. Our model is trained in two steps: **first**, to utilize non-differentiable compression quality information, we train a differentiable compression-quality-based ranker network to predict scores related to compression quality factors. The CQR is a Siamese network with pair-wise inputs, which have same content but different compression degrees, shown in Figure 1. **Second**, based on the previous CQR, we design the specific quality-related loss and feature matching loss to train the de-blocking network for JPEG artifacts removal. The de-blocking network training strategy is shown in Figure 2. Below we will detail the architecture, training strategy and loss functions.

3.1 Compression Quality Ranker

Network architecture. The architecture of the CQR is depicted in Figure 1. Inspired by the methods [Saquil *et al.*, 2018; Zhang *et al.*, 2019a], we use a Siamese network to predict the relative scores of compression quality factors. Specifically, this network contains two same branches that share weights and each branch is composed of convolutional layers, batch normalization, Leaky ReLU, global average pooling and fully connected layers. To update parameters, we send the output scores to the margin loss module to learn the compression-quality ranking order of the pair-wise input images, whose compression quality factors are different. The preparation of the pair-wise input images and the optimization loss function of CQR are described below.

Compression quality paired images. First, the clean image is compressed by setting different JPEG-compression quality factors (QF), which range from a to b. Note that a lower QF indicates a higher compression degree, which results in poorer image quality and more obvious artifacts.

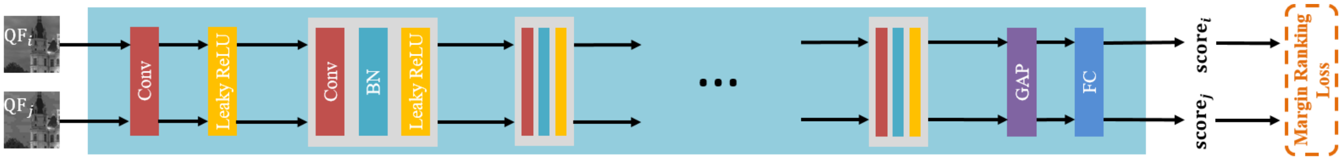


Figure 1: Step1: Training Compression Quality Ranker (CQR). During training, the ranker is a Siamese network with paired images of different compression factors as inputs. The Margin ranking loss, i.e., Eq. (2) is used to constrain the output score, so that the size relationship between the scores is consistent with the size relationship between quality factors. Since the ranker is differentiable, the output scores that highly related to compression quality factors can be used to optimize the subsequent de-blocking network.

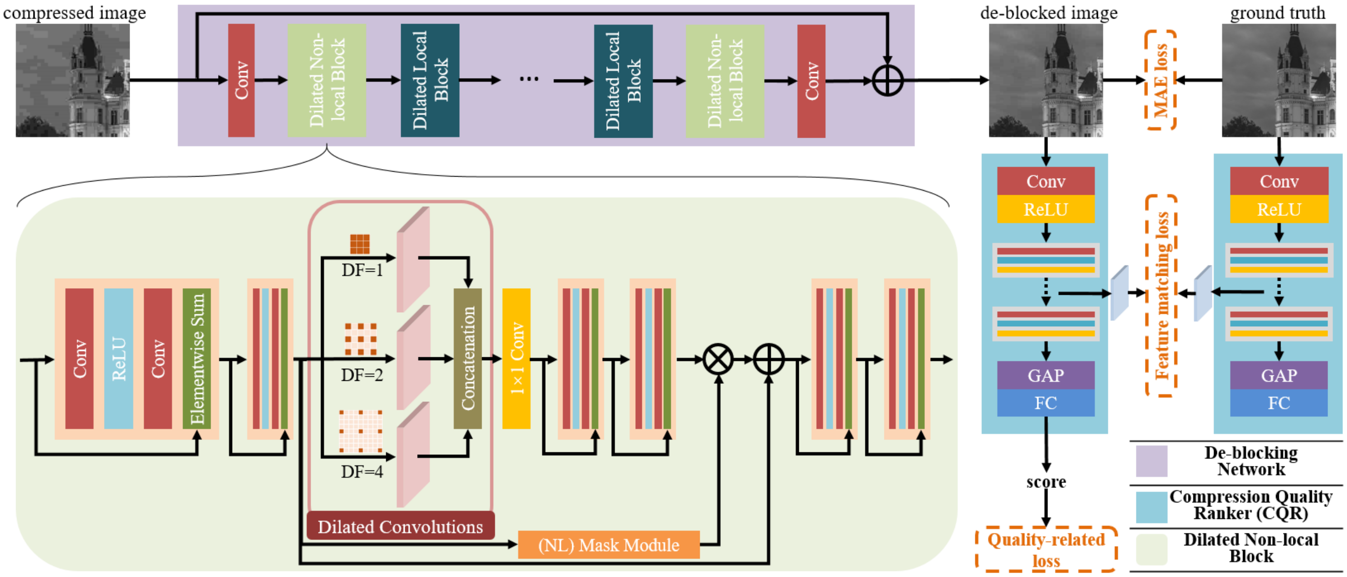


Figure 2: Step2: Training De-blocking Network. Based on the CQR (blue box), we train the de-blocking network (purple box). The de-blocking network is composed of dilated non-local blocks and dilated local blocks. Dilated non-local block consists of residual blocks, non-local mask module and dilated convolutions. By applying different dilated factors, dilated convolutions can extract multi-scale features. Therefore, the single network is able to handle multiple compression qualities. During training, we fix the network parameters of the well-trained CQR. Then we use the combined loss, i.e., Eq. (6), to train the de-blocking network.

Then, we divide the range of QF into n levels, and their corresponding labels are set to $1, 2, \cdots, n$. For example, the image with $[a, a + (b - a)/n]$ range of QF is labeled as "1". Obviously, a higher label reflects a better image quality. During the training procedure, we randomly select two compressed images which have same content but different labels as the pair-wise input images to optimize the ranker.

Margin ranking loss. After obtaining the paired input images, the CQR predicts their two scores respectively:

$$\begin{aligned} s_i &= CQR(x_i), \\ s_j &= CQR(x_j), \end{aligned} \tag{1}$$

where s_i and s_j represent the output scores of image x_i and image x_j, respectively. To train the ranker $CQR(\cdot)$, we adopt margin ranking loss as the optimization loss function, which is commonly used in ranking problems [Zhang *et al.*, 2019a] and can be expressed as:

$$\begin{aligned} L_M &= max\left\{0, (s_i - s_j) * \eta + \varepsilon\right\}, \\ &\begin{cases} \eta = 1, & if\ \ b_i < b_j \\ \eta = -1, & if\ \ b_i > b_j \end{cases}, \end{aligned} \tag{2}$$

where b_i and b_j represent the labels of image x_i and image x_j, respectively. Obviously, if the relationship between s_i and s_j is consistent with the relationship between b_i and b_j, the loss equals 0. Based on the well-trained CQR, a higher output score indicates a higher image quality.

3.2 De-blocking Network

Network architecture. The architecture of the de-blocking network is shown in the purple box in Figure 2. As can be seen, the network consists of two basic blocks, i.e., dilated non-local block and dilated local block. Inspired by RNAN [Zhang *et al.*, 2019b], we directly quote its non-local mask module to extract global context information. The difference between these two blocks is that the non-local operation is utilized in dilated non-local blocks to generate the mask map. Therefore, we mainly introduce the dilated non-local block. It consists of dilated convolutions and residual blocks with the skip connection operation. The residual block is composed of convolutional layer and ReLU. Then we describe the multi-scale dilated convolutions. RNAN [Zhang *et al.*, 2019b] and other methods [Zhang *et al.*, 2017; Fan *et* *al.*, 2018] mostly need to train individual models for different compression quality factors, which limits their practical values. To handle this obstacle, we introduce dilated convolutions to extract multi-scale features. As shown in Figure 2, different dilated factors (DF) can expand the filter to different scales and increase receptive fields. By using concatenation operation, we can obtain multi-scale representations, which allows a single model to achieve the ability to solve the multiple compression quality issue. Additionally, the dilated convolutions bring no extra parameters.

Training strategy. The training strategy of the de-blocking network is depicted in Figure 2.As mentioned above, the output score of well-trained CQR can effectively represent the compression quality. Therefore, we use the scores to guide the training of de-blocking network. Firstly, the compressed image is processed by de-blocking network to generate the de-blocked image. This de-blocked image and the corresponding ground truth are used to formulate mean absolute error, which represents pixel-level information. Secondly, the de-blocked image and ground truth are sent into well-trained CQR. The CQR output score of de-blocked image formulates quality-related loss, which measures the quality degree of de-blocked image. Additionally, we introduce feature matching loss to measure the feature-level similarity between de-blocked image and ground truth. Due to the three loss functions and dilated convolutions, the network can be efficiently optimized both in quantitative and qualitative measurements, and the single model can handle multiple compression qualities. The specific loss functions are described below.

3.3 Loss Functions

MAE loss. In this paper, we use mean absolute error to constrain the pixel-level similarity. MAE loss is expressed as:

$$L_{MAE} = \|M(x) - y\|_1 \,, \tag{3}$$

where x and y represent the compressed image and ground truth, respectively. $M(\cdot)$ represents the mapping function of de-blocking network.

Quality-related loss. As mentioned above, the larger score output by the well-trained CQR indicates a better image quality. Therefore, we can utilize CQR to measure the quality of the de-blocked image. Since the proposed ranker can be seen as a classification process, which is the most fundamental semantic-related vision task [Liu *et al.*, 2019], it is able to provide semantic-level information to guide the de-blocking network training. This loss can be presented as:

$$L_Q = sigmoid(-CQR(M(x))), \tag{4}$$

where $CQR(M(x))$ is the score of the de-blocked image $M(x)$. To formulate a loss function, we take the negative value of the score, and perform a sigmoid operation to make the loss range from 0 to 1.

Feature matching loss. Perceptual loss is widely applied in various image restoration tasks using a pre-trained VGG network [Simonyan and Zisserman, 2014]. However, the VGG model is not trained to remove JPEG artifacts. Therefore, it cannot provide the JPEG artifacts-related features well. Unlike the traditional perceptual loss, our feature matching loss is formulated by several feature maps extracted from CQR, which can more accurately reflect JPEG artifacts information. Feature matching loss can be presented as:

$$L_F = \mathbb{E}\left[\sum_i \frac{1}{N_i}\left\|CQR^{(i)}(M(x)) - CQR^{(i)}(y)\right\|_1\right], \tag{5}$$

where $CQR^{(i)}(\cdot)$ represents the feature map extracted from the ith layer of CQR, and N_i is the element number.

Overall loss. The above three loss functions constitute the overall loss to train the de-blocking network:

$$L = L_{MAE} + \alpha L_Q + \beta L_F, \tag{6}$$

where α and β are used to balance different loss functions.

4 Experiments

4.1 Experimental Datasets and Settings

Datasets. In our experiments, we use four datasets: DIV2K [Agustsson and Timofte, 2017], BSDS500 [Arbelaez *et al.*, 2010], LIVE1 [Sheikh, 2005] and Classic5 [Zeyde *et al.*, 2010]. The DIV2K dataset (800 images) is used for training, while the other three datasets are used for testing. As mentioned above, our model is trained in a stage-wise fashion: first CQR is trained and frozen, and then the de-blocking network is optimized. Note that the whole procedure is only implemented on the Y channel image of YCrCb space. The specific settings of training and testing are described below.

Training settings of CQR. To obtain pair-wise input images for CQR, we use the standard JPEG compression scheme to obtain compressed images with different JPEG quality factors. The factors range from 1 to 30 and we divide the range into 3 levels to label compressed images. Specifically, the images within range $[1, 10]$, $[11, 20]$, and $[21, 30]$ are labeled as 1, 2 and 3, respectively. Then, we randomly select two compressed images with the same content but different labels to form a pair-wise training data.

Training settings of de-blocking network. We generate the compressed images by using three JPEG quality factors, which are 10, 20 and 30. Then, for each ground truth image, there are 3 corresponding compressed images with different quality factors. Other methods mostly use 3 quality factors to train 3 models, and each model can only handle specific quality factor. As mentioned above, our single model is able to handle multiple JPEG qualities. Thus, we use all the 3 quality factors to train our de-blocking network. The parameters α and β are empirically set as 0.01 and 0.9, respectively.

Testing settings of de-blocking network. We use the same compression operation on BSDS500 (500 images), LIVE1 (29 images) and Classic5 (5 images). They are compressed with 3 quality factors: 10, 20 and 30. We use these compressed images to evaluate the performance of our model both in different datasets and in different quality factors.

4.2 Experimental Details

Evaluation metrics. For quantitative assessment, we use PSNR, structural similarity (SSIM) and PSNR-B [Yim and Bovik, 2010] to evaluate de-blocking performance. Note that the PSNR-B is specifically designed and more sensitive to blocking artifacts than SSIM.

Quality	w/o CQR	w/o DC	w/o L_Q	w/o L_F	Ours
10	29.21|0.801|28.81	29.38|0.792|28.92	29.41|0.802|28.98	29.32|0.780|28.90	**29.42**|**0.807**|**29.01**
20	31.41|0.827|30.89	30.77|0.799|30.12	31.87|0.851|31.23	31.71|0.809|31.13	**31.97**|**0.865**|**31.29**
30	32.77|0.889|31.90	31.19|0.832|30.27	32.79|0.890|31.90	32.26|0.861|31.39	**32.91**|**0.891**|**31.98**

Table 1: Quantitative Ablation Study on PSNR|SSIM|PSNR-B Values. We evaluate the effect of the components. "Ours" means all 4 components are used. The dataset used in this experiment is BSDS500. The best results are boldfaced.

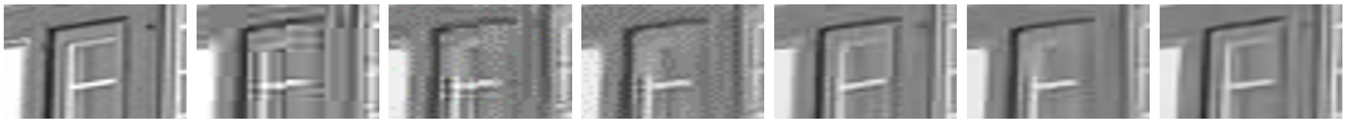
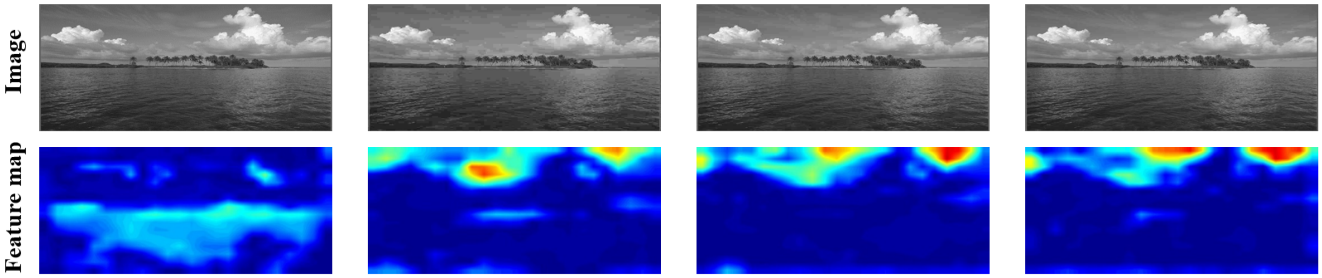

(a) GT (b) JPEG (c) w\o CQR (d) w\o DC (e) w\o L_Q (f) w\o L_F (g) Ours

Figure 3: Qualitative Ablation Study. We evaluate the visual effect of the components. "Ours" means all 4 components are used. The testing images is from LIVE1 dataset with JPEG quality factor 10.



(a) GT (b) QF=10 (c) QF=20 (d) QF=30

Figure 4: Visualization of the feature maps output by our ranker. As can be seen, our ranker has a low response to the clean image and high responses to JPEG compressed images. We enlarge the feature maps to make them correspond to the spatial position of the images.

Implementation details. As for the network architecture, the compression quality ranker (CQR) utilizes the VGG network [Simonyan and Zisserman, 2014], which includes 10 convolutional layers. The specific dilated factors are set to 1, 2 and 4. The ε in margin ranking loss is set to 0.5. For optimizations, we use Adam optimizer [Kingma and Ba, 2014] to train our model. The learning rate is initialized to 10^{-4} and decreased to half every 1×10^5 iterations. We implement the proposed model with two Titan Xp GPUs by using PyTorch.

4.3 Ablation Study

Our model contains 4 important components: Compression Quality Ranker (CQR), dilated convolutions (DC), quality-related loss (L_Q) and feature matching loss (L_F), which allow our model to achieve promising performance both in qualitative and quantitative evaluation. We perform ablation experiments on the above 4 components to analyze their effects, which are shown in the Table 1 and Figure 3.

Effect of Compression Quality Ranker. Obviously, when removing the CQR module, L_Q and L_F no longer exist. As shown in the Table 1, the PSNR-B of "w/o CQR" decreases 0.20 at $q = 10$, but only decreases 0.08 at $q = 30$, which indicates that, by introducing compression quality information, CQR can guide the network to effectively remove severe compression artifacts.

Effect of dilated convolutions. As shown in Table 1, when removing DC, the network recovers better at $q = 10$, but fails to achieve good performance at $q = 20$ and 30. Without DC, the multi-scale context information cannot be extracted, thus the details of the images are difficult to be recovered. The "w/o DC" can only extract single-scale features, which cannot be used to handle multiple compression qualities.

Effect of quality-related loss. In Table 1, the PSNR-B metric of "w/o L_Q" is 0.03, 0.06, 0.08 lower than ours at all the three qualities, which indicates that L_Q can improve network performance. To prove that our ranker is able to discriminate different quality factors, we visualize the learned feature maps in Figure 4. Obviously, our ranker has a low response to the clean image and high responses to JPEG compressed images, especially in the area where artifacts are noticeable. Additionally, for different quality factors, the ranker generates different feature maps, which can provide discriminative information to guide the subsequent de-blocking network training.

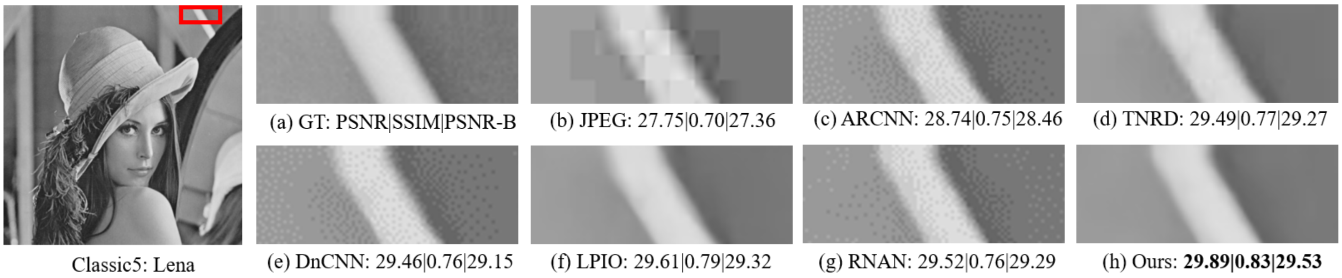
Effect of feature matching loss. When L_F is removed, the visual quality metric SSIM drops heavily, due to the feature-level constraint disappears. The network cannot be effectively optimized at the visual level. In addition, the PSNR and PSNR-B also decreased, which indicates that L_F reflects more valuable information compared with L_Q.

4.4 Comparison with the State-of-the-Art Methods

We compare our method with state-of-the-art LD [Li *et al.*, 2014], ARCNN [Dong *et al.*, 2015], TNRD [Chen and Pock,

Dataset	Quality	LD [2014]	ARCNN [2015]	TNRD [2017]	DnCNN [2017]	LPIO [2018]	RNAN [2019b]	Ours
Classic5	10	28.39|0.800|27.59	29.03|0.793|28.78	29.28|0.799|29.04	29.40|0.803|29.10	29.35|0.801|29.04	**29.94**|0.809| 29.67	29.92|**0.837**|**29.71**
	20	30.30|0.858|29.98	31.15|0.852|30.60	31.47|0.858|31.05	31.63|0.861|31.19	31.58|0.856|31.12	32.14|0.863|31.78	**32.89**|**0.886**|**32.41**
	30	31.50|0.882|31.31	32.51|0.881|32.00	32.78|0.884|32.24	32.91|0.886|32.36	32.86|0.883|32.28	**33.38**|0.890|32.81	33.31|**0.893**|**32.97**
LIVE1	10	28.26|0.805|27.68	28.96|0.808|28.77	29.15|0.811|28.88	29.19|0.812|28.91	29.17|0.811|28.89	29.60|**0.819**|29.26	**29.81**|0.810|**29.59**
	20	30.19|0.871|30.08	31.29|0.873|30.79	31.46|0.877|31.04	31.59|0.880|31.08	31.52|0.876|31.07	**32.11**|0.876|31.74	32.04|**0.891**|**31.77**
	30	31.32|0.898|31.27	32.67|0.904|32.22	32.84|0.906|32.28	32.98|0.909|32.35	32.99|0.907|32.31	33.43|0.907|32.69	**33.57**|**0.912**|**32.81**
BSDS500	10	28.03|0.782|27.29	28.56|0.783|28.54	28.42|0.781|28.30	28.84|0.783|28.44	28.81|0.781|28.39	29.15|0.787|28.89	**29.41**|**0.801**|**29.03**
	20	29.82|0.851|29.57	30.42|0.852|30.39	30.35|0.854|30.16	31.05|0.857|30.29	30.92|0.855|30.07	31.37|0.861|30.76	**31.87**|**0.868**|**31.08**
	30	30.89|0.883|30.83	31.51|0.884|31.47	31.36|0.887|31.12	32.36|0.891|31.43	32.31|0.886|31.27	32.79|0.890|31.90	**32.91**|**0.901**|**32.00**

Table 2: Quantitative comparisons on PSNR|SSIM|PSNR-B values. The best results are boldfaced. Note that, unlike other methods, our method trains single model to handle different quality factors.



Classic5: Lena

(a) GT: PSNR|SSIM|PSNR-B

(b) JPEG: 27.75|0.70|27.36

(c) ARCNN: 28.74|0.75|28.46

(d) TNRD: 29.49|0.77|29.27

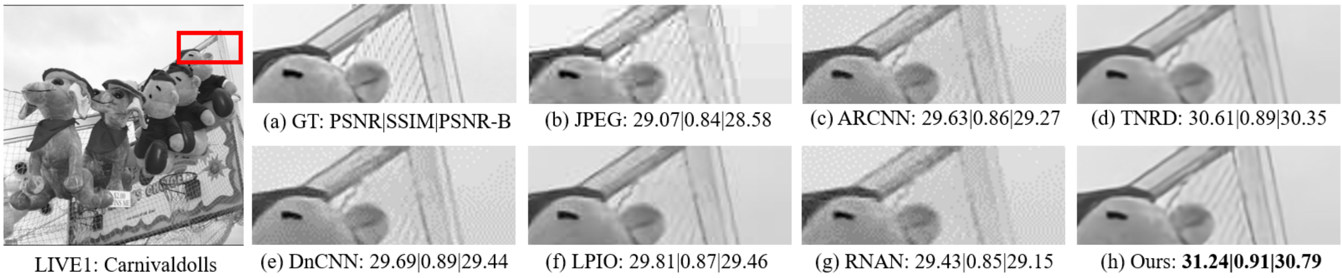
(e) DnCNN: 29.46|0.76|29.15

(f) LPIO: 29.61|0.79|29.32

(g) RNAN: 29.52|0.76|29.29

(h) Ours: **29.89|0.83|29.53**

Figure 5: Qualitative comparison on the state-of-the-art methods. The testing image is from Classic5 dataset with JPEG quality factor 10.



LIVE1: Carnivaldolls

(a) GT: PSNR|SSIM|PSNR-B

(b) JPEG: 29.07|0.84|28.58

(c) ARCNN: 29.63|0.86|29.27

(d) TNRD: 30.61|0.89|30.35

(e) DnCNN: 29.69|0.89|29.44

(f) LPIO: 29.81|0.87|29.46

(g) RNAN: 29.43|0.85|29.15

(h) Ours: **31.24|0.91|30.79**

Figure 6: Qualitative Comparison on the state-of-the-art methods. The testing image is from LIVE1 dataset with JPEG quality factor 20.

2017], DnCNN [Zhang *et al.*, 2017], LPIO [Fan *et al.*, 2018] and RNAN [Zhang *et al.*, 2019b]. As shown in Table 2, Figures 5 and 6, our model achieves comparable or superior performance both in qualitative and quantitative measurements.

Quantitative evaluation. As shown in Table 2, we show PSNR, SSIM and PSNR-B values to evaluation our proposed model with other methods. As we can see, compared with other methods, our method achieves the best overall performance in all the three metrics. Since the PSNR-B is specifically designed for de-blocking problems, the higher PSNB-B of our method indicates our model is more suitable for this JPEG artifacts removal task.

Qualitative evaluation. As shown in Figures 5 and 6, our model is able to simultaneously achieve compression artifacts removal and details restoration. Specifically, in smooth areas, blocking artifacts are well removed. Moreover, in the areas with complex textures, such as the edge in Figure 5 and the lines in Figure 6, our de-blocked images contain more fluent edges and no additional blurring or ringing effects are introduced. Especially, in Figure 6, the lines are very close to the ground truth.

5 Conclusion

In this paper, we propose compression quality ranker-guided networks for JPEG artifacts removal. Specifically, a compression quality ranker is introduced to measure different compression degrees. Based on the differentiable ranker, we then design quality-related loss and feature matching loss to guide de-blocking network optimization. Additionally, dilated convolutions are introduced to extract multi-scale features, which allows our single model to handle multiple JPEG qualities. Experiments show that our method achieves satisfactory performance in both qualitative and quantitative quality.

Acknowledgments

This work was supported by the National Key R&D Program of China under Grant 2017YFB1300201, the National Natural Science Foundation of China (NSFC) under Grants 61901433, U19B2038 and 61620106009 as well as the Fundamental Research Funds for the Central Universities under Grant WK2100100030.

References

[Agustsson and Timofte, 2017] E. Agustsson and R. Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *CVPR Workshops*, 2017.

[Arbelaez *et al.*, 2010] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2010.

[Chang *et al.*, 2013] H. Chang, M. K Ng, and T. Zeng. Reducing artifacts in jpeg decompression via a learned dictionary. *IEEE Transactions on Signal Processing*, 62(3):718–728, 2013.

[Chen and Pock, 2017] Y. Chen and T. Pock. A flexible framework for fast and effective image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1256–1272, 2017.

[Dong *et al.*, 2015] C. Dong, Y. Deng, C. Change Loy, and X. Tang. Compression artifacts reduction by a deep convolutional network. In *ICCV*, pages 576–584, 2015.

[Fan *et al.*, 2018] Q. Fan, D. Chen, L. Yuan, G. Hua, N. Yu, and B. Chen. Decouple learning for parameterized image operators. In *ECCV*, pages 442–458, 2018.

[Foi *et al.*, 2007] 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*, 16(5):1395–1411, 2007.

[Fu *et al.*, 2019] X. Fu, Z.-J. Zha, F. Wu, X. Ding, and J. Paisley. Jpeg artifacts reduction via deep convolutional sparse coding. In *ICCV*, pages 2501–2510, 2019.

[Goodfellow *et al.*, 2014] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014.

[Guo and Chao, 2017] J. Guo and H. Chao. One-to-many network for visually pleasing compression artifacts reduction. In *CVPR*, pages 3038–3047, 2017.

[He *et al.*, 2016] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[Jancsary *et al.*, 2012] J. Jancsary, S. Nowozin, and C. Rother. Loss-specific training of non-parametric image restoration models: A new state of the art. In *ECCV*, pages 112–125, 2012.

[Johnson *et al.*, 2016] J. Johnson, A. Alahi, and F.-F. Li. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711, 2016.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Li *et al.*, 2014] Y. Li, F. Guo, R. T Tan, and M. S Brown. A contrast enhancement framework with jpeg artifacts suppression. In *ECCV*, pages 174–188, 2014.

[Li *et al.*, 2017] T. Li, X. He, L. Qing, Q. Teng, and H. Chen. An iterative framework of cascaded deblocking and super-resolution for compressed images. *IEEE Transactions on Multimedia*, 20(6):1305–1320, 2017.

[Liu *et al.*, 2018] D. Liu, B. Wen, X. Liu, Z. Wang, and T. S Huang. When image denoising meets high-level vision tasks: a deep learning approach. In *IJCAI*, 2018.

[Liu *et al.*, 2019] D. Liu, H. Zhang, and Z. Xiong. On the classification-distortion-perception tradeoff. In *NeurIPS*, pages 1204–1213, 2019.

[Saquil *et al.*, 2018] Y. Saquil, K. I. Kim, and P. Hall. Ranking cgans: Subjective control over semantic image attributes. *arXiv preprint arXiv:1804.04082*, 2018.

[Sheikh, 2005] HR Sheikh. Live image quality assessment database release 2. *http://live. ece. utexas. edu/research/quality*, 2005.

[Simonyan and Zisserman, 2014] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[Vandal *et al.*, 2018] T. Vandal, E. Kodra, S. Ganguly, A. R Michaelis, R. R Nemani, and A. R Ganguly. Generating high resolution climate change projections through single image super-resolution: An abridged version. In *IJCAI*, 2018.

[Wang *et al.*, 2016] Z. Wang, D. Liu, S. Chang, Q. Ling, Y. Yang, and T. S Huang. D^3: Deep dual-domain based fast restoration of jpeg-compressed images. In *CVPR*, 2016.

[Wang *et al.*, 2019] N. Wang, J. Li, L. Zhang, and B. Du. Musical: multi-scale image contextual attention learning for inpainting. In *IJCAI*, pages 3748–3754, 2019.

[Yim and Bovik, 2010] C. Yim and A. C. Bovik. Quality assessment of deblocked images. *IEEE Transactions on Image Processing*, 20(1):88–98, 2010.

[Zeyde *et al.*, 2010] R. Zeyde, M. Elad, and M. Protter. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, 2010.

[Zhang *et al.*, 2017] 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*, 26(7):3142–3155, 2017.

[Zhang *et al.*, 2018] R. Zhang, S. Tang, L. Liu, Y. Zhang, J. Li, and S. Yan. High resolution feature recovering for accelerating urban scene parsing. In *IJCAI*, 2018.

[Zhang *et al.*, 2019a] W. Zhang, Y. Liu, C. Dong, and Y. Qiao. RanksrGan: Generative adversarial networks with ranker for image super-resolution. In *ICCV*, 2019.

[Zhang *et al.*, 2019b] Y. Zhang, K. Li, K. Li, B. Zhong, and Y. Fu. Residual non-local attention networks for image restoration. In *ICLR*, 2019.

[Zhu *et al.*, 2018] H. Zhu, X. Peng, V. Chandrasekhar, L. Li, and J.-H. Lim. Dehazegan: When image dehazing meets differential programming. In *IJCAI*, pages 1234–1240, 2018.