Towards Flexible Blind JPEG Artifacts Removal

Jiaxi Jiang   Kai Zhang[*]   Radu Timofte

Computer Vision Lab, ETH Zurich, Switzerland

jiaxijiang@student.ethz.ch   {kai.zhang, timofter}@vision.ee.ethz.ch

https://github.com/jiaxi-jiang/FBCNN

Abstract

Training a single deep blind model to handle different quality factors for JPEG image artifacts removal has been attracting considerable attention due to its convenience for practical usage. However, existing deep blind methods usually directly reconstruct the image without predicting the quality factor, thus lacking the flexibility to control the output as the non-blind methods. To remedy this problem, in this paper, we propose a flexible blind convolutional neural network, namely FBCNN, that can predict the adjustable quality factor to control the trade-off between artifacts removal and details preservation. Specifically, FBCNN decouples the quality factor from the JPEG image via a decoupler module and then embeds the predicted quality factor into the subsequent reconstructor module through a quality factor attention block for flexible control. Besides, We find existing methods are prone to fail on non-aligned double JPEG images even with only one pixel shift, and we thus propose a double JPEG degradation model to augment the training data. Extensive experiments on single JPEG images, more general double JPEG images and real-world JPEG images demonstrate that our proposed FBCNN achieves favorable performance against state-of-the-art methods in terms of both quantitative metrics and visual quality.

1. Introduction

JPEG [39] is one of the most widely-used image compression algorithms and formats due to its simplicity and fast encoding/decoding speeds. JPEG compression splits an image into 8×8 blocks and applies discrete cosine transform (DCT) to each block. The DCT coefficients are then divided by a quantization table and rounded to the nearest integer. The elements in the quantization table control the compression ratio and the rounding operation is the only lossy operation in the whole process. The quantization table is usually represented by an integer called quality factor (QF) ranging from 0 to 100, where a lower quality factor means less storage size but more lost information. Inspired by the success of deep neural networks (DNNs) for image classification [25, 37], researchers began to resort to DNNs for JPEG artifacts removal and have achieved notable academic success.

However, existing methods for JPEG artifacts removal generally have four limitations in real applications: (1) Most existing DNNs based methods [6, 7, 11, 28, 54] trained a specific model for each quality factor, lacking the flexibility to learn a single model for different JPEG quality factors. (2) DCT based methods [13, 17, 53] need to obtain the DCT coefficients or quantization table as input, which is only stored in JPEG format. Besides, when images are compressed multiple times, only the most recent compression information is stored. (3) To solve the first problem, some recent work [13, 15, 50] resort to training a single model for a large range of quality factors. However, these blind methods can only provide a deterministic reconstruction result for each input, ignoring the need for user preferences. (4) Existing methods are all trained with synthetic images which assumes that the low-quality images are compressed only once. However, most images from the Internet are compressed multiple times. Despite some progresses for real recompressed images, *e.g.*, from Twitter [11, 15], a detailed and complete study on double JPEG artifacts removal is still missing.

To tackle the above problems, we design a flexible blind convolutional neural network, namely FBCNN, for real JPEG image restoration. Our FBCNN is a single model that can deal with JPEG images with different quality factors. In addition, FBCNN can work independent of the image formats, as it directly processes images in pixel-domain, without the need to access the metadata of images. By further decoupling the latent quality factor from the input JPEG image, we can use this important parameter to guide the artifacts removal process. As a controllable variable with clear physical meaning, the predicted quality factor can also be adjusted via interactive selection to achieve a balance

---

[*]Corresponding author.

between artifacts removal and details preservation. To address the problems with real-world JPEG images, we provide a detailed study on the restoration of images with double JPEG compression. We find that existing blind methods are prone to fail when the 8×8 blocks of double JPEG compression are not aligned and $\text{QF}_1 \leq \text{QF}_2$. However, our quality factor predictor can help to explain the behavior of current blind methods under unseen scenarios. We provide comprehensive empirical evidence showing that blind methods work are easy to be misled by the unseen compound artifacts, resulting in an unpleasant reconstructed output. By correcting the predicted quality factor, FBCNN instead can boost the performance on complex double JPEG images. To obtain a fully blind model, we further propose two solutions: correcting QF to the smaller one which can be estimated by our dominant QF estimation method or augmenting the training data with non-aligned double JPEG images.

To summarize, the main contributions of this paper are:

(1) A flexible blind convolutional neural network for JPEG artifacts removal (FBCNN) is proposed. FBCNN can predict the latent quality factor to guide the image restoration. The predicted quality factor can be adjusted manually to control the preference between artifacts removal and details preservation according to the user's needs.

(2) We perform a thorough analysis of double JPEG images and provide solutions to take a step towards the restoration of real images. To the best of our knowledge, this is the first attempt to handle double non-aligned JPEG compression. We hope that the community will gradually begin to consider this more challenging and realistic scenario.

(3) We demonstrate the effectiveness of FBCNN on synthetic and real JPEG images with complex degradation settings. Our proposed FBCNN provides a useful solution for practical applications.

2. Related Work

JPEG Artifacts Removal Networks. Learning-based methods have made notable progress in JPEG artifacts removal in the past few years. Dong *et al*. [11] first introduced deep learning to remove JPEG artifacts, inspired by the success of super-resolution network [12]. Zhang *et al*. [50] employed batch normalization [22] and residual learning [20] strategies to speed up the training process and boost the performance on general blind image restoration tasks. A wavelet transform based network was presented in [28] as the generalization of dilated convolution [46] and subsampling, leading to a large improvement. Fu *et al*. [15] proposed a deep convolutional sparse coding network that combines model-based methods with deep learning. Besides, dual-domain convolutional network based methods [17, 23, 53, 55] were proposed to take advantage of redundancies on both pixel and DCT domains. Recently, Ehrlich *et al*. [13] trained their networks with the utilization

of quantization table as prior information, which allows a single model to correct artifacts at any quality factor and achieved state-of-the-art results.

Double JPEG Compression. Double JPEG compression has been studied in the area of image forensics for a long time, as detection of double compression can provide important clues for the recovery of image processing history. Fu *et al*. [14] showed that if an image has been JPEG compressed only once, then the first digits of the quantized JPEG coefficients follow a Benford-like logarithmic law. In [3, 4, 8, 29], double JPEG compression was classified into two cases: aligned and non-aligned. Chen *et al*. [8] formulated the periodic characteristics of JPEG images in both spatial and DCT domains and showed that such periodic characteristics will be changed after recompression. Recently, learning-based methods [2, 31, 40] were proposed to detect double JPEG compression. The estimation of the first quantization table of JPEG images is also a challenging problem and studied in both aligned [16, 33, 44, 47] and non-aligned cases [5, 10, 45]. However, these methods focus on analyzing the DCT coefficients, which are only stored in JPEG format. Besides, the research on double JPEG compression restoration is still missing.

Flexible Image Restoration. Flexible image generation based on the conditional variable has drawn much attention in *e.g*. text-to-image generation [26, 35, 43] and facial attribute editing [9, 21, 27]. However, these methods can not be directly adopted in image restoration. Zhang *et al*. [51] proposed to take a tunable noise level map as the input to handle noise on different levels. In [52], a PCA-based dimensionality stretching of the degradation parameters was proposed to take blur kernel and noise level as input for super-resolution. Wang *et al*. [41] proposed a novel controllable framework for interactive image restoration. He *et al*. [19] focused on the images with multiple degradations and added the multi-dimensional degradation information as input. These methods usually assume that the controllable variable is provided, but such information is almost unknown in real applications. This encourages us to work towards a flexible blind solution for image restoration.

3. Proposed Method

In this section, we first introduce the architecture of our FBCNN, and then present its advantage over other state-of-the-art methods, especially for practical recompressed JPEG images.

3.1. Flexible Blind Artifacts Removal Network

Fig. 1 illustrates the overall architecture of our proposed FBCNN. FBCNN is an end-to-end model which takes a
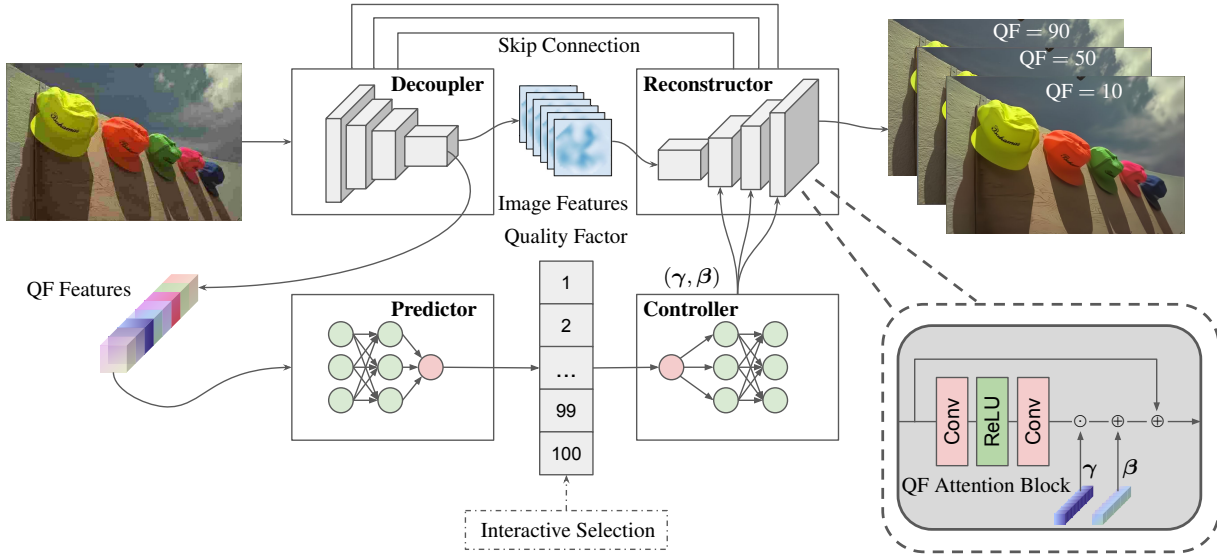
Figure 1. The architecture of the proposed FBCNN for JPEG artifacts removal. FBCNN consists of four parts, *i.e.*, decoupler, quality factor predictor, flexible controller, and image reconstructor. The decoupler extracts the deep features from the input corrupted JPEG image and then splits them into image features and QF features which are subsequently fed into the reconstructor and predictor, respectively. The controller gets the estimated QF from the predictor and then generates QF embeddings. The QF attention block enables the controller to make the reconstructor produce different results according to different QF embeddings. The predicted quality factor can be changed with interactive selections to have a balance between artifacts removal and details preservation.

JPEG image as input and directly generates the output image. Specifically, FBCNN consists of four components: decoupler, QF predictor, flexible controller, and image reconstructor. The network is fairly straightforward, with each component designed to achieve a specific task.

Decoupler: The decoupler aims to extract the deep features and decouple the latent quality factor from the input image. It involves four scales, each of which has an identity skip connection to the reconstructor. Residual blocks are adopted in each scale, and each residual block is composed of two 3×3 convolution layers with ReLU activation in the middle. 2×2 strided convolutions are adopted for the downscaling operations. The number of output channels in each layer from the first to the fourth scale is set to 64, 128, 256, 512, respectively. The decoupler is shared by image reconstruction and quality factor prediction branches. The image branch processes the encoded deep image features with 512 channels. We set the number of residual blocks in each scale as 4. The output of the image branch is fed into the reconstructor. The quality factor branches first use residual blocks to extract higher-level information. Because the quality factor is a single value for each image, we adopt a global average pooling layer to get the global information from the image features.

Quality Factor Predictor: The QF predictor is a 3-layer MLP (multilayer perceptron) that takes as input the 512-dimensional QF features and produces an estimated quality factor $\mathrm{QF_{est}}$ of the compressed image. We set the number of nodes in each hidden layer as 512 for a better prediction. During training, patches with small sizes may only include limited information and correspond to multiple quality factors so that the quality factor can not be accurately estimated, which may lead to an unstable training process. Therefore, we use the L1 loss function to avoid too much penalty for such outliers. Let N be the batch size during training, the loss for quality factor estimation in each batch can be written as:

$$\mathcal{L}_{\mathrm{QF}} = \frac{1}{N} \sum_{i=1}^{N} \left\| \mathrm{QF}_{\mathrm{est}}^i - \mathrm{QF}_{\mathrm{gt}}^i \right\|_1 . \tag{1}$$

Flexible Controller: The flexible controller is a 4-layer MLP and takes as input the quality factor, representing the degree of compression of the targeted image. The controller aims to learn an embedding of the given quality factor that can be fused into the reconstructor for flexible control. Inspired by recent research in spatial feature transform [32, 42], the controller learns a mapping function that outputs a modulation parameter pair $(\boldsymbol{\gamma}, \boldsymbol{\beta})$ which embeds the given quality factor. Specifically, the first three layers

of MLP generate shared intermediate conditions, which are then split into three parts corresponding to the three scales in the reconstructor. In the last layer of MLP, we learn different parameter pairs for different scales in reconstructor whereas shared (γ, β) are broadcasted to the QF Attention block within the same scale.

Image Reconstructor: The image reconstructor includes three scales and receives image features from decoupler and quality factor embedding parameters (γ, β) to generate the restored clean image. The QF attention block is an important component of the reconstructor. The number of QF attention blocks in each scale is set to 4. The learned parameter pair (γ, β) adaptively influences the outputs by applying an affine transformation spatially to each intermediate feature map inside the QF attention block of each scale.

After obtaining (γ, β) from the controller, the transformation is carried out by scaling and shifting feature maps of a specific layer:

$$\boldsymbol{F}_{\text{out}} = (1 + \boldsymbol{\gamma}) \odot \boldsymbol{F}_{\text{in}} \oplus \boldsymbol{\beta}, \tag{2}$$

where $\boldsymbol{F}_{\text{in}}$ and $\boldsymbol{F}_{\text{out}}$ denote the feature maps before and after the affine transformation, and $\odot$ is referred to as element-wise multiplication.

Given N training samples within a batch, the goal of the image reconstructor is to minimize the following L1 loss function between reconstructed image $\mathbf{I}_{\text{rec}}$ and the original ground-truth image $\mathbf{I}_{\text{gt}}$:

$$\mathcal{L}_{\text{rec}} = \frac{1}{N} \sum_{i=1}^{N} \left\| \mathbf{I}_{\text{rec}}^{i} - \mathbf{I}_{\text{gt}}^{i} \right\|_{1} . \tag{3}$$

Overall, the complete training objective can be written as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{rec}} + \lambda \cdot \mathcal{L}_{\text{QF}}, \tag{4}$$

where λ controls the balance between image reconstruction and QF estimation.

3.2. Comparison with Other Design Choices

In the following, we will clarify the differences between the proposed FBCNN and two alternative design choices.

A blind model without QF prediction: Existing blind methods only provide a deterministic result, ignoring the need of the user's preference. Besides, as we will discuss in Sec. 3.3, although the pure blind model performs favorably for single JPEG artifacts removal without knowing the quality factor, it does not generalize well to real corrupted images whose artifacts are more complex. FBCNN can be viewed as multiple deblockers and can control the trade-off between JPEG artifacts removal and details preservation.

Cascaded QF prediction and non-blind model: It is also possible to design a QF predictor cascaded by a non-blind method like CBD-Net [18]. However, our method enjoys some benefits compared with such a cascaded design: First, for accurate quality factor estimation, a convolutional network starting from the same scale as the input image is needed, which would increase the total model size and cost more training and inference time. Instead, we only add a relatively small prediction branch. Second, our decoupler shared parameters for QF estimation and image reconstruction, accelerating the convergence of predicting QF. On the contrary, in cascaded design, inaccurate QF estimation would lead to an unstable training process. It might be a solution to train a QF predictor and then freeze it to train the second part for reconstruction. Nevertheless, it would cost more training time than our joint training schedule. Fourth, in cascaded networks, the predicted parameter is treated as the input of the second part and propagates through the whole encoder-decoder architecture. Instead, our predicted parameter QF is the only input to the decoder part. We can change the QF to adjust different outputs during inference without the need to change the encoded image features, which saves half of the inference time.

3.3. Restoration of Double JPEG Images

Limitations of existing methods: Although some existing work claimed to work on recompressed JPEG images, a detailed study on the restoration of double JPEG compression is still missing. We find that the current blind methods always fail when the blocks of two JPEG compression are not aligned and $\text{QF}_1 \leq \text{QF}_2$, even if there is an only one-pixel shift between two compression.

Let us look at an example in Fig. 2, where the appearances of JPEG images with different compression settings can be observed. To get non-aligned double JPEG images, we remove the first row and the first column of the image between the first compression with QF_1 and the second one with QF_2. For aligned double JPEG with QF = (90, 10), (10, 90), and non-aligned double JPEG with QF = (90, 10)*, the blocking effects are similar to single compression with QF = 10: the edges of 8×8 blocks are apparent. However, in the case of non-aligned double JPEG with QF = (10, 90)*, the blocking edges are not clear anymore. We test representative blind methods DnCNN [50] and QGAC [13] on these images.

As shown in Fig. 2, in cases of QF = 90, 10, (90, 10), the blocking effects are well removed by both methods. DnCNN also works well on QF = (10, 90), while QGAC fails in this case because QGAC extracts the quantization table from the JPEG image, but JPEG images only keep the most recent compression information. Therefore, we conclude that existing quantization table-based methods are not suitable for real application.
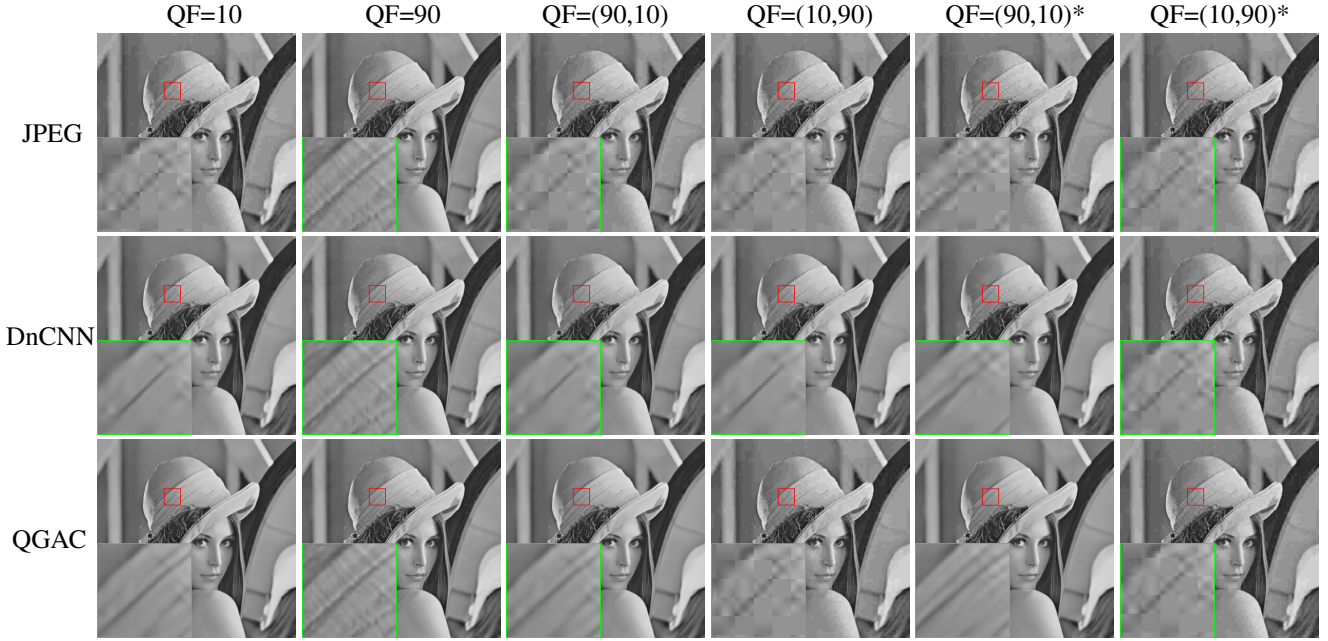
| QF=10 | QF=90 | QF=(90,10) | QF=(10,90) | QF=(90,10)* | QF=(10,90)* |

JPEG

DnCNN

QGAC

Figure 2. Visual comparisons of a JPEG image with different degradation settings and their restored results by DnCNN and QGAC. QF = (QF_1, QF_2) denotes that the image is firstly compressed with QF_1 and then compressed with QF_2. '*' means there is a pixel shift (1,1) between blocks of two compression. Even only a shift of one pixel between two compression can lead to failures of existing methods.

However, in the case of non-aligned double JPEG compression when $QF_1 = 10$ and $QF_2 = 90$, both methods do not work. Since our FBCNN is also a pixel-based blind method like DnCNN but can predict the quality factor, it can be used to explain the behavior behind a blind method. We test FBCNN using the same images. Not surprisingly, we get a similar, almost unchanged reconstructed result, but we find the predicted quality factor is 90. We continue to test other images with non-aligned double JPEG compression and $QF_1 < QF_2$, finding that the predicted quality factor is always close to QF_2. This is to say, blind methods trained with single JPEG compression image pairs are always misled by the appearance of non-aligned double JPEG images with $QF_1 < QF_2$. They also do not work when $QF_1 = QF_2$.

In summary, we classify double JPEG compression into two categories: simple and complex compression. Simple compression corresponds to non-aligned double JPEG with $QF_1 > QF_2$ and all aligned double JPEG compression, which is actually equivalent to single JPEG compression. Complex compression corresponds to non-aligned double JPEG with $QF_1 \leq QF_2$, where composite artifacts occur. We test images with these degradation settings by a recent double JPEG compression algorithm [31], finding that only images with non-aligned double JPEG with $QF_1 \leq QF_2$ is identified as double JPEG compression, which further support our arguments.

To overcome the problem with non-aligned double JPEG compression, we propose two solutions, from the perspectives of adjusting the QF to utilize our flexible network and augmenting the training data.

FBCNN trained with a single JPEG degradation model with dominant QF correction: Since our FBCNN can provide different outputs by setting different quality factors, correcting the predicted QF to the smaller one, which actually dominates the main compression, is expected to improve the restoration results. However, to get a fully blind model, it is crucial to infer the smaller quality factor automatically. By utilizing the property of JPEG compression, we find that the quality factor of a JPEG image with single compression can be obtained by doing another JPEG compression with all possible QFs. The image's QF corresponds to the global minimum of the MSE (mean squared error) between two JPEG images. We further extend this method to challenging non-aligned double JPEG images with $QF_1 < QF_2$. We apply another JPEG compression with all possible QFs after a shift in the range of 0 to 7 in two directions. We also calculate the MSE curves for each shift possibility between the two JPEG images. For each MSE curve, we search for the first minimum. It can be found that among all first minimum, the QF at the smallest first minimum is always close to QF_1, while the QF at the global minimum is approximate to QF_2. Besides, we constrain the MSE of the smallest first minimum to be smaller than a threshold T to have more robust results. We empirically set T to 30 in our experiment. We name the FBCNN model with dominant QF correction as FBCNN-D.

FBCNN trained with double JPEG degradation model: We can also solve this problem by augmenting the training data using images with double JPEG compression. We pro-

Table 1. PSNR|SSIM|PSNRB results of different methods on **grayscale** JPEG images with **single** compression. Please note that the methods marked with * train a specific model for each quality factor. The best two results are highlighted in red and blue colors, respectively.

Dataset	Quality	JPEG	ARCNN*	MWCNN*	DnCNN	DCSC	QGAC	FBCNN (Ours)
Classic5	10	27.82|0.760|25.21	29.03|0.793|28.76	30.01|0.820|29.59	29.40|0.803|29.13	29.62|0.810|29.30	29.84|0.812|29.43	30.12|0.822|29.80
	20	30.12|0.834|27.50	31.15|0.852|30.59	32.16|0.870|31.52	31.63|0.861|31.19	31.81|0.864|31.34	31.98|0.869|31.37	32.31|0.872|31.74
	30	31.48|0.867|28.94	32.51|0.881|31.98	33.43|0.893|32.62	32.91|0.886|32.38	33.06|0.888|32.49	33.22|0.892|32.42	33.54|0.894|32.78
	40	32.43|0.885|29.92	33.32|0.895|32.79	34.27|0.906|33.35	33.77|0.900|33.23	33.87|0.902|33.30	34.05|0.905|33.12	34.35|0.907|33.48
LIVE1	10	27.77|0.773|25.33	28.96|0.808|28.68	29.69|0.825|29.32	29.19|0.812|28.90	29.34|0.818|29.01	29.51|0.825|29.13	29.75|0.827|29.40
	20	30.07|0.851|27.57	31.29|0.873|30.76	32.04|0.889|31.51	31.59|0.880|31.07	31.70|0.883|31.18	31.83|0.888|31.25	32.13|0.889|31.57
	30	31.41|0.885|28.92	32.67|0.904|32.14	33.45|0.915|32.80	32.98|0.909|32.34	33.07|0.911|32.43	33.20|0.914|32.47	33.54|0.916|32.83
	40	32.35|0.904|29.96	33.61|0.920|33.11	34.45|0.930|33.78	33.96|0.925|33.28	34.02|0.926|33.36	34.16|0.929|33.36	34.53|0.931|33.74
BSDS500	10	27.80|0.768|25.10	29.10|0.804|28.73	29.61|0.820|29.14	29.21|0.809|28.80	29.32|0.813|28.91	29.46|0.821|28.97	29.67|0.821|29.22
	20	30.05|0.849|27.22	31.28|0.870|30.55	31.92|0.885|31.15	31.53|0.878|30.79	31.63|0.880|30.92	31.73|0.884|30.93	32.00|0.885|31.19
	30	31.37|0.884|28.53	32.67|0.902|31.94	33.30|0.912|32.34	32.90|0.907|31.97	32.99|0.908|32.08	33.07|0.912|32.04	33.37|0.913|32.32
	40	32.30|0.903|29.49	33.55|0.918|32.78	34.27|0.928|33.19	33.85|0.923|32.80	33.92|0.924|32.92	34.01|0.927|32.81	34.33|0.928|33.10

pose a new degradation model to synthesize the non-aligned double JPEG image $\mathbf{y}$ from the uncompressed image $\mathbf{x}$ via

$$\mathbf{y} = \text{JPEG}(\text{shift}(\text{JPEG}(\mathbf{x}, \text{QF}_1)), \text{QF}_2). \tag{5}$$

For shift operation, we randomly remove the first i rows and j columns of the image after the first compression, where $0 \leq i, j \leq 7$. When trained with double JPEG compressed images, the weight of quality factor loss is set to zero. Then the dominant quality factor can be trained in an unsupervised way. We name the FBCNN model with augmented training data as FBCNN-A. Note that our double JPEG degradation model can also be applied to other tasks such as blind single image super-resolution [49].

4. Experiments

4.1. Data Preparation and Network Training

For fair comparisons, JPEG images used during training and evaluation are all generated by the MATLAB JPEG encoder. We use the Y channel of YCbCr space for grayscale image comparison, and the RGB channels for color image comparison. Following [13], we employ DIV2K [1] and Flickr2K [38] as our training data. During training, we randomly extract patch pairs with the size 128×128, and the quality factor is randomly sampled from 10 to 95. We set λ to 0.1. To optimize the parameters of FBCNN, we adopt the Adam solver [24] with batch size 256. The learning rate starts from 1×10^{-4} and decays by a factor of 0.5 every 4×10^4 iterations and finally ends with 1.25×10^{-5}. We train our model with PyTorch on eight NVIDIA GeForce GTX 2080Ti GPUs. It takes about two days to obtain FBCNN.

4.2. Single JPEG Image Restoration

Grayscale JPEG image restoration We first evaluate the performance of the proposed FBCNN on images with single JPEG compression. We test on the commonly used benchmarks: Classic5 [48], LIVE1 [36] and the test set of BSDS500 [30]. We compare our proposed FBCNN with

ARCNN [11], MWCNN [28], DnCNN [50], DCSC [15], QGAC [13]. It should be pointed out that ARCNN and MWCNN train a single network for each specific value of quality factor, and DCSC is trained with quality factors from 10 to 40. Only DnCNN, QGAC, and our FBCNN cover a full range of quality factors. We calculate the PSNR, SSIM, and PSNR-B for quantitative assessment. The quantitative results are shown in Table 1. Our method has significantly better results than other blind methods and moderately better results than MWCNN, which trains each model for a specific quality factor. For subjective comparisons, some restored images of different approaches on the LIVE1 dataset have been presented. As can be seen in Fig. 3, the results of our FBCNN are more visually pleasing.

Color JPEG image restoration We also train our model on RGB channels, referred to as FBCNN-C. We compare FBCNN-C with QGAC, which is a state-of-the-art method, especially for color JPEG image restoration. The evaluation is made on LIVE1 [36], testset of BSDS500 [30], and ICB [34] dataset. Although QGAC is specially designed for color JPEG image artifacts removal, we still get better performance by setting the input/output channels as 3. The result is shown in Table 2.

Table 2. PSNR|SSIM|PSNRB results of QGAC and FBCNN-C on **color** JPEG images with **single** compression.

Dataset	QF	JPEG	QGAC	FBCNN-C (Ours)
LIVE1	10	25.69|0.743|24.20	27.62|0.804|27.43	27.77|0.803|27.51
	20	28.06|0.826|26.49	29.88|0.868|29.56	30.11|0.868|29.70
	30	29.37|0.861|27.84	31.17|0.896|30.77	31.43|0.897|30.92
	40	30.28|0.882|28.84	32.05|0.912|31.61	32.34|0.913|31.80
BSDS500	10	25.84|0.741|24.13	27.74|0.802|27.47	27.85|0.799|27.52
	20	28.21|0.827|26.37	30.01|0.869|29.53	30.14|0.867|29.56
	30	29.57|0.865|27.72	31.33|0.898|30.70	31.45|0.897|30.72
	40	30.52|0.887|28.69	32.25|0.915|31.50	32.36|0.913|31.52
ICB	10	29.44|0.757|28.53	32.06|0.816|32.04	32.18|0.815|32.15
	20	32.01|0.806|31.11	34.13|0.843|34.10	34.38|0.844|34.34
	30	33.20|0.831|32.35	35.07|0.857|35.02	35.41|0.857|35.35
	40	33.95|0.840|33.14	32.25|0.915|31.50	36.02|0.866|35.95

(a) JPEG (29.64dB)   (b) ARCNN (31.15dB)   (c) MWCNN (32.38dB)   (d) DnCNN (31.36dB)

(e) DCSC (31.68dB)   (f) QGAC (31.97dB)   (g) FBCNN (32.51dB)   (h) Ground Truth

Figure 3. Visual comparisons of different methods on a **single** JPEG image with QF = 10 (BSDS500: 14088).

(a) JPEG (27.45dB)   (b) QF = 10 (28.13dB)   (c) QF = 30 (29.34dB)   (d) QF = 90 (28.05dB)

Figure 4. An example to show the flexibility of FBCNN by setting different QFs into the network. The JEPG image is 'LIVE1: cemetry' compressed with quality factor 30. Although the artifacts around the words can be effectively removed when the set QF is small, the texture on the bricks becomes blurred. Users can get the desired results according to their preference through interactive selection by FBCNN.

Table 3. PSNR|SSIM|PSNRB results of different methods on **grayscale** JPEG images with **non-aligned double** compression. The testing images are synthesized from the LIVE1 dataset. The best two results are highlighted in red and blue colors, respectively.

Type	QF	JPEG	DnCNN	DCSC	QGAC	FBCNN (Ours)	FBCNN-D (Ours)	FBCNN-A (Ours)
	(30,10)	27.49|0.762|25.62	28.95|0.805|28.61	29.08|0.810|28.81	29.24|0.818|28.94	29.46|0.820|29.11	29.46|0.820|29.10	29.44|0.818|29.12
QF$_1$>QF$_2$	(50,10)	27.65|0.769|25.69	29.13|0.810|28.76	29.25|0.815|28.96	29.42|0.823|29.08	29.64|0.825|29.23	29.65|0.825|29.22	29.61|0.823|29.20
	(50,30)	30.62|0.866|28.85	32.20|0.895|31.50	32.30|0.897|31.78	32.32|0.899|31.72	32.61|0.902|31.88	32.61|0.902|31.89	32.69|0.901|32.24
	(10,10)	26.48|0.715|25.08	27.73|0.765|27.49	27.76|0.768|27.59	27.78|0.771|27.59	27.96|0.774|27.75	27.95|0.774|27.74	29.25|0.777|28.14
QF$_1$=QF$_2$	(30,30)	29.98|0.847|28.53	31.40|0.878|30.86	31.48|0.880|31.10	31.43|0.881|30.99	31.64|0.884|31.14	31.65|0.884|31.14	31.94|0.886|31.73
	(50,50)	31.58|0.888|30.18	33.12|0.912|32.44	33.28|0.914|32.80	33.28|0.914|32.50	33.38|0.917|32.61	33.45|0.914|32.85	33.70|0.919|33.34
	(10,30)	27.55|0.760|26.94	28.33|0.790|28.17	28.31|0.789|28.19	28.30|0.791|28.18	28.29|0.791|28.15	28.94|0.802|28.82	29.38|0.816|29.30
QF$_1$<QF$_2$	(10,50)	27.69|0.768|27.41	28.30|0.791|28.24	28.40|0.794|28.35	28.23|0.791|28.18	28.20|0.789|28.14	28.96|0.801|28.88	29.52|0.820|29.45
	(30,50)	30.61|0.865|29.60	31.89|0.890|31.46	32.08|0.893|31.78	31.81|0.891|31.43	31.96|0.893|31.50	32.31|0.895|31.94	32.64|0.900|32.49

Flexible JPEG image restoration To demonstrate the flexibility of FBCNN, we show an example in Fig. 4. By setting different quality factors, we can get results with different perception qualities. Users can make an interactive selection according to their preferences.

4.3. Double JPEG Image Restoration

The focus of our paper is to remove the complex double JPEG compression artifacts, which is an important step towards real image restoration. So we also evaluate the performance of current state-of-the-art methods and our pro-

posed methods on images with double JPEG compression. We compare our methods with blind methods: DnCNN, DCSC, QGAC. The comparison is conducted using different combinations of quality factors (QF$_1$, QF$_2$) on the LIVE1 dataset. Each original image is JPEG compressed with QF$_1$, cropped by a random shift (4, 4) to the upper left corner, and JPEG compressed with QF$_2$.

The numerical and visual results are reported in Table 3 and Fig. 5. As shown in Table 3, when changing the order of QF$_1$ and QF$_2$, although the differences between the PSNR values of JPEG images are generally smaller than
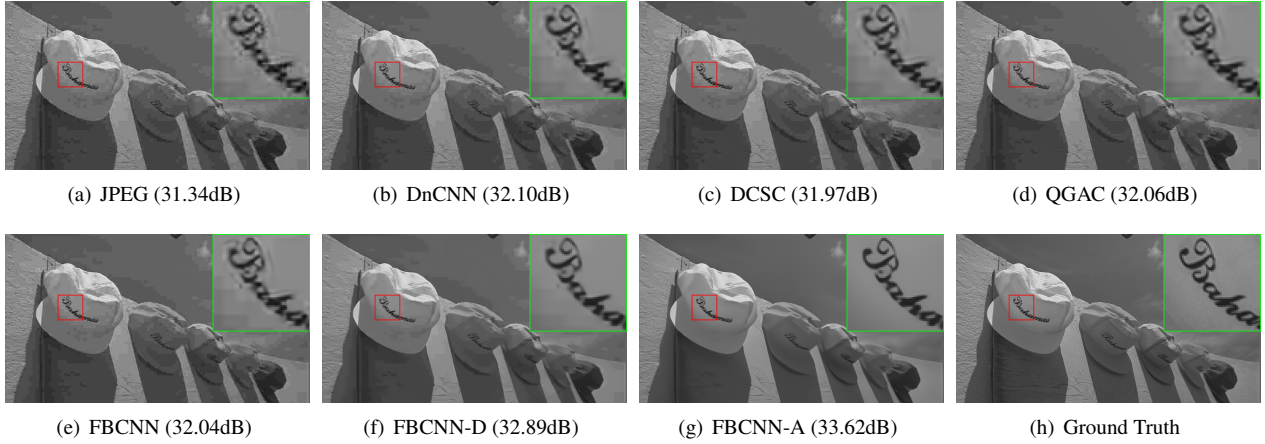
(a) JPEG (31.34dB) (b) DnCNN (32.10dB) (c) DCSC (31.97dB) (d) QGAC (32.06dB)

(e) FBCNN (32.04dB) (f) FBCNN-D (32.89dB) (g) FBCNN-A (33.62dB) (h) Ground Truth

Figure 5. Visual comparisons of image 'LIVE1: caps' with **non-aligned double** JPEG compression. This image is degraded by the first JPEG with $QF_1 = 10$, pixel shift = (4, 4), the second JPEG with $QF_2 = 30$ successively.
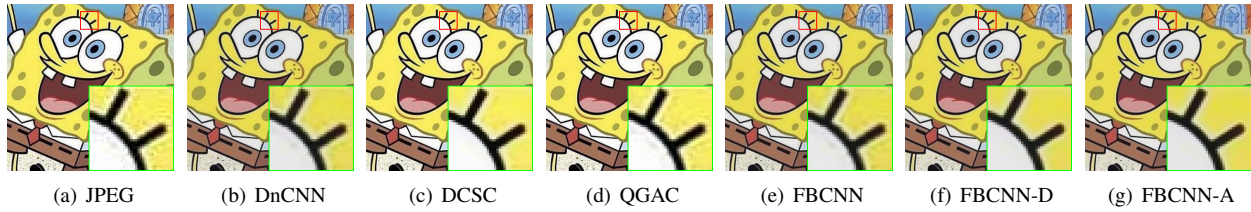

(a) JPEG (b) DnCNN (c) DCSC (d) QGAC (e) FBCNN (f) FBCNN-D (g) FBCNN-A

Figure 6. Visual comparisons of an example from our Meme dataset.

0.05 dB, a significant drop in performance can be seen on other methods and our FBCNN. Since DCSC is only trained with small quality factors from 10 to 40, it generally performs better than DnCNN, QGAC, and FBCNN when $QF_1 < QF_2$. Despite some benefits for double JPEG compression, it should be pointed out that it is not reasonable to use a model trained with low quality factors to tackle all kinds of JPEG images. When dealing with relatively high-quality images, it tends to give more blurry results.

We also examine the effectiveness of our proposed two solutions to non-aligned double JPEG restoration. FBCNN-D is obtained based on FBCNN by correcting the quality factor by dominant QF estimation during inference. FBCNN-A is obtained by augmenting the training data with our proposed double JPEG degradation model. Table 3 shows that by correcting the predicted quality factor, FBCNN-D largely improves the PSNR when $QF_1 < QF_2$. FBCNN-A further improves performance when $QF_1 < QF_2$. The difficult case when $QF_1 = QF_2$ also sees an improvement on FBCNN-A.

4.4. Real-World JPEG Image Restoration

Besides the above experiments on synthetic test images, we also conduct experiments on real images to demonstrate the effectiveness of the proposed FBCNN. We collect 400 meme images from the Internet, as this kind of image is of-ten compressed many times. Fig. 6 shows a test example on our collected Meme dataset. Since there are no ground-truth high-quality images and no reliable no-reference image quality assessment (IQA) metrics, we do not report the quantitative results. We leave the study of no-reference IQA for JPEG compression artifacts removal for future works.

5. Conclusions

In this paper, we proposed a flexible blind JPEG artifacts removal network (FBCNN) for real JPEG image restoration. FBCNN decouples the quality factor from the input image via a decoupler and then embeds the predicted quality factor into the subsequent reconstructor through a quality factor attention block for flexible control. The predicted quality factor can also be adjusted to achieve a balance between artifacts removal and details preservation. Besides, we address non-aligned double JPEG restoration tasks to take steps towards real JPEG images with severe degradations. Extensive experiments on single JPEG images, the more general double JPEG images, and real-world JPEG images demonstrate the flexibility, effectiveness, and generalizability of our proposed FBCNN for restoring different kinds of degraded JPEG images.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 126–135, 2017. 6

[2] Mauro Barni, Luca Bondi, Nicolò Bonettini, Paolo Bestagini, Andrea Costanzo, Marco Maggini, Benedetta Tondi, and Stefano Tubaro. Aligned and non-aligned double jpeg detection using convolutional neural networks. *Journal of Visual Communication and Image Representation*, 49:153–163, 2017. 2

[3] Mauro Barni, Andrea Costanzo, and Lara Sabatini. Identification of cut & paste tampering by means of double-jpeg detection and image segmentation. In *International Symposium on Circuits and Systems*, pages 1687–1690. IEEE, 2010. 2

[4] Tiziano Bianchi and Alessandro Piva. Analysis of non-aligned double jpeg artifacts for the localization of image forgeries. In *International Workshop on Information Forensics and Security*, pages 1–6. IEEE, 2011. 2

[5] Tiziano Bianchi and Alessandro Piva. Image forgery localization via block-grained analysis of jpeg artifacts. *IEEE Transactions on Information Forensics and Security*, 7(3):1003–1017, 2012. 2

[6] Lukas Cavigelli, Pascal Hager, and Luca Benini. Cas-cnn: A deep convolutional neural network for image compression artifact suppression. In *International Joint Conference on Neural Networks*, pages 752–759. IEEE, 2017. 1

[7] Yunjin Chen and Thomas Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1256–1272, 2016. 1

[8] Yi-Lei Chen and Chiou-Ting Hsu. Detecting recompression of jpeg images via periodicity analysis of compression artifacts for tampering detection. *IEEE Transactions on Information Forensics and Security*, 6(2):396–406, 2011. 2

[9] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8789–8797, 2018. 2

[10] Nandita Dalmia and Manish Okade. Robust first quantization matrix estimation based on filtering of recompression artifacts for non-aligned double compressed jpeg images. *Signal Processing: Image Communication*, 61:9–20, 2018. 2

[11] Chao Dong, Yubin Deng, Chen Change Loy, and Xiaoou Tang. Compression artifacts reduction by a deep convolutional network. In *International Conference on Computer Vision*, pages 576–584, 2015. 1, 2, 6

[12] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *European Conference on Computer Vision*, pages 184–199. Springer, 2014. 2

[13] Max Ehrlich, Ser-Nam Lim, Larry Davis, and Abhinav Shrivastava. Quantization guided jpeg artifact correction. In *European Conference on Computer Vision*, 2020. 1, 2, 4, 6

[14] Dongdong Fu, Yun Q Shi, and Wei Su. A generalized benford's law for jpeg coefficients and its applications in image forensics. In *Security, Steganography, and Watermarking of Multimedia Contents IX*, volume 6505, page 65051L. International Society for Optics and Photonics, 2007. 2

[15] Xueyang Fu, Zheng-Jun Zha, Feng Wu, Xinghao Ding, and John Paisley. Jpeg artifacts reduction via deep convolutional sparse coding. In *International Conference on Computer Vision*, pages 2501–2510, 2019. 1, 2, 6

[16] Fausto Galvan, Giovanni Puglisi, Arcangelo Ranieri Bruna, and Sebastiano Battiato. First quantization matrix estimation from double compressed jpeg images. *IEEE Transactions on Information Forensics and Security*, 9(8):1299–1310, 2014. 2

[17] Jun Guo and Hongyang Chao. Building dual-domain representations for compression artifacts reduction. In *European Conference on Computer Vision*, pages 628–644. Springer, 2016. 1, 2

[18] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1712–1722, 2019. 4

[19] Jingwen He, Chao Dong, and Yu Qiao. Interactive multi-dimension modulation with dynamic controllable residual learning for image restoration. In *European Conference on Computer Vision*. Springer, 2020. 2

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 2

[21] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. Attgan: Facial attribute editing by only changing what you want. *IEEE Transactions on Image Processing*, 28(11):5464–5478, 2019. 2

[22] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456. PMLR, 2015. 2

[23] Yoonsik Kim, Jae Woong Soh, and Nam Ik Cho. Agarnet: Adaptively gated jpeg compression artifacts removal network for a wide range quality factor. *IEEE Access*, 8:20160–20170, 2020. 2

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 6

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems*, 25:1097–1105, 2012. 1

[26] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip HS Torr. Controllable text-to-image generation. In *Neural Information Processing Systems*, 2019. 2

[27] Ming Liu, Yukang Ding, Min Xia, Xiao Liu, Errui Ding, Wangmeng Zuo, and Shilei Wen. Stgan: A unified selective transfer network for arbitrary image attribute editing. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3673–3682, 2019. 2

[28] Pengju Liu, Hongzhi Zhang, Kai Zhang, Liang Lin, and Wangmeng Zuo. Multi-level wavelet-cnn for image restora-

tion. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2018. 1, 2, 6

[29] Weiqi Luo, Zhenhua Qu, Jiwu Huang, and Guoping Qiu. A novel method for detecting cropped and recompressed image block. In *International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages II–217. IEEE, 2007. 2

[30] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *International Conference on Computer Vision*, volume 2, pages 416–423. IEEE, 2001. 6

[31] Jinseok Park, Donghyeon Cho, Wonhyuk Ahn, and Heung-Kyu Lee. Double jpeg detection in mixed jpeg quality factors using deep convolutional neural network. In *European Conference on Computer Vision*, pages 636–652, 2018. 2, 5

[32] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019. 3

[33] Cecilia Pasquini, Giulia Boato, and Fernando Pérez-González. Multiple jpeg compression detection by means of benford-fourier coefficients. In *International Workshop on Information Forensics and Security*, pages 113–118. IEEE, 2014. 2

[34] Rawzor. Image compression benchmark. 6

[35] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International Conference on Machine Learning*, pages 1060–1069. PMLR, 2016. 2

[36] HR Sheikh. Live image quality assessment database release 2. *http://live. ece. utexas. edu/research/quality*, 2005. 6

[37] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, May 2015. 1

[38] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang. Ntire 2017 challenge on single image super-resolution: Methods and results. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 114–125, 2017. 6

[39] Gregory K Wallace. The jpeg still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1):xviii–xxxiv, 1992. 1

[40] Qing Wang and Rong Zhang. Double jpeg compression forensics based on a convolutional neural network. *EURASIP Journal on Information Security*, 2016(1):1–12, 2016. 2

[41] Wei Wang, Ruiming Guo, Yapeng Tian, and Wenming Yang. Cfsnet: Toward a controllable feature space for image restoration. In *International Conference on Computer Vision*, pages 4140–4149, 2019. 2

[42] Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 606–615, 2018. 3

[43] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1316–1324, 2018. 2

[44] Fei Xue, Ziyi Ye, Wei Lu, Hongmei Liu, and Bin Li. Mse period based estimation of first quantization step in double compressed jpeg images. *Signal Processing: Image Communication*, 57:76–83, 2017. 2

[45] Heng Yao, Hongbin Wei, Chuan Qin, and Xinpeng Zhang. An improved first quantization matrix estimation for non-aligned double compressed jpeg images. *Signal Processing*, 170:107430, 2020. 2

[46] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *International Conference on Learning Representations*, 2016. 2

[47] Liyang Yu, Qi Han, Xiamu Niu, SM Yiu, Junbin Fang, and Ye Zhang. An improved parameter estimation scheme for image modification detection based on dct coefficient analysis. *Forensic Science International*, 259:200–209, 2016. 2

[48] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *International Conference on Curves and Surfaces*, pages 711–730. Springer, 2010. 6

[49] Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. Designing a practical degradation model for deep blind image super-resolution. In *IEEE Conference on International Conference on Computer Vision*, 2021. 6

[50] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017. 1, 2, 4, 6

[51] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, 2018. 2

[52] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Learning a single convolutional super-resolution network for multiple degradations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3262–3271, 2018. 2

[53] Xiaoshuai Zhang, Wenhan Yang, Yueyu Hu, and Jiaying Liu. Dmcnn: Dual-domain multi-scale convolutional neural network for compression artifacts removal. In *International Conference on Image Processing*, pages 390–394. IEEE, 2018. 1, 2

[54] Y Zhang, K Li, K Li, B Zhong, and Y Fu. Residual non-local attention networks for image restoration. In *International Conference on Learning Representations*, 2019. 1

[55] Bolun Zheng, Yaowu Chen, Xiang Tian, Fan Zhou, and Xuesong Liu. Implicit dual-domain convolutional network for robust color image compression artifact reduction. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019. 2