IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 30, 2021, accepted December 21, 2021, date of publication December 23, 2021,
date of current version January 6, 2022.

Digital Object Identifier 10.1109/ACCESS.2021.3138106

WCDGAN: Weakly Connected Dense Generative Adversarial Network for Artifact Removal of Highly Compressed Images

BINGHUA XIE, HAO ZHANG, AND CHEOLKON JUNG, (Member, IEEE)
School of Electronic Engineering, Xidian University, Xian 710071, China

Corresponding author: Cheolkon Jung (zhengzk@xidian.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61872280.

ABSTRACT In highly compressed images, i.e. quality factor $q \leq 10$, JPEG compression causes severe compression artifacts including blocking, banding, ringing and color distortion. The compression artifacts seriously degrade image quality, which is not conducive to subsequent tasks, such as object detection and semantic segmentation. In this paper, we propose a weakly connected dense generative adversarial network for artifacts removal of highly compressed images, named WCDGAN. WCDGAN has three main ingredients of mixed convolution, weakly connected dense block (WCDB), and mixed attention. In the loss function, we add a perceptual loss to generate photo-realistic images with compression artifact removal. Experimental results show that WCDGAN successfully removes compression artifacts and produces sharp edges, clear textures and vivid colors even in highly compressed images. Moreover, WCDGAN outperforms state-of-the-art methods for compression artifact removal in terms of peak signal-to-noise ratio (PSNR) and structural similarity (SSIM).

INDEX TERMS Image compression, convolutional neural network, generative adversarial network, weak connection, attention mechanism, dilated convolution.

I. INTRODUCTION

JPEG [1] is a widely used image format that represents rich and vivid contents due to the sophisticated compression algorithm. It first converts an RGB image into YCbCr color space and downsamples chroma components. Then, it performs the discrete cosine transform (DCT) [2] on luma and chroma components in the form of non-overlapping 8×8 blocks. Given a quality factor, the corresponding quantization tables (QTs) are obtained, and the DCT coefficients are quantized based on the QT. Finally, the quantized DCT coefficients are converted into bitstream for transmission according to the encoding rules. However, it can be seen from Eq. (1) that due to the floor function, the DCT coefficients at the encoding end and the DCT coefficients at the decoding end are different.

$$D_{de} = \left\lfloor \frac{D_{en}}{Q} \right\rfloor \times Q \tag{1}$$

The associate editor coordinating the review of this manuscript and approving it for publication was Ioannis Schizas.

where $\lfloor \rfloor$ is the floor operation, Q is the quantization step length, D_{en} and D_{de} denote DCT coefficient at the encoding end and DCT coefficient at the decoding end, respectively.

Thus, the compression distortion is caused by quantization. As shown in Eqs. (2)-(4), DCT coefficients represent information in the frequency domain and the distortion of DCT coefficients affects the pixel values involved in the calculation in the spatial domain. Moreover, the DCT transformation is carried out block by block and thus the compressed images contain blocking and banding artifacts.

$$F(u, v) = c(u)c(v) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i, j) \cos A \cos B \tag{2}$$

$$A = \frac{(i + 0.5)\pi}{N} u \tag{3}$$

$$B = \frac{(j + 0.5)\pi}{N} v \tag{4}$$

where generally $N = 8$; i and j represent horizontal and vertical coordinates in the spatial domain, respectively; u and v represent frequency; F represents DCT coefficient; f represents pixel values in the spatial domain. Note that when

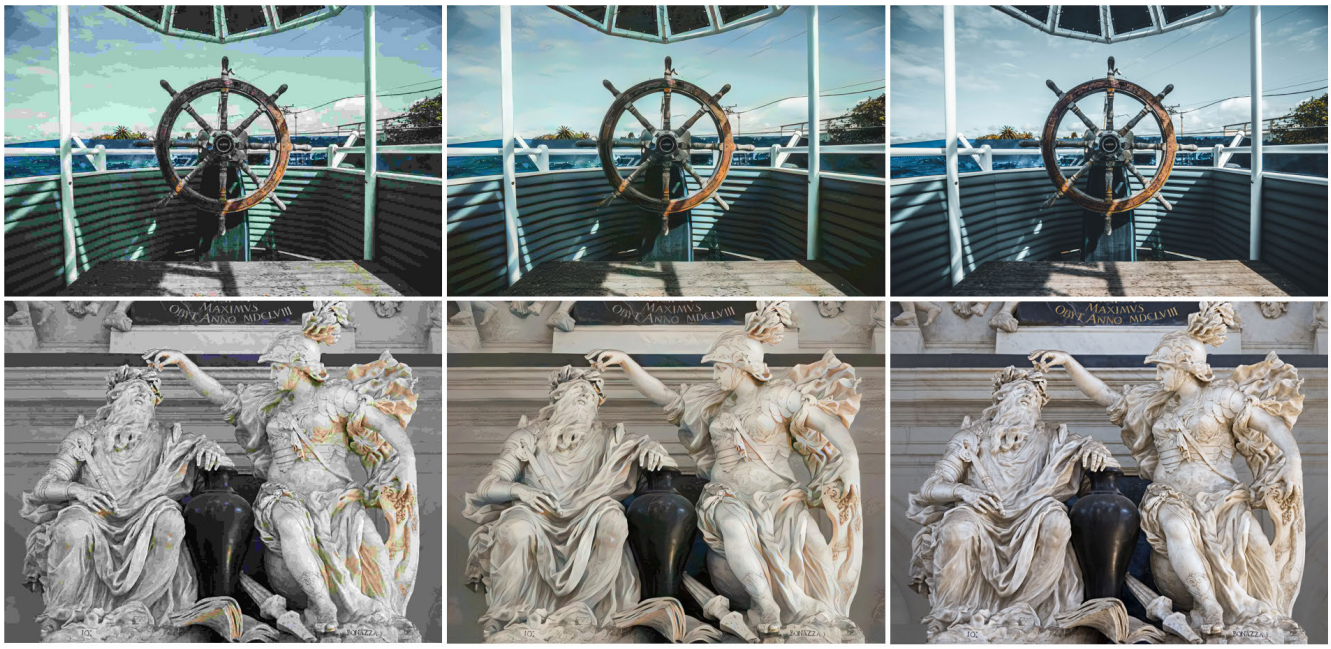


FIGURE 1. Compression artifact removal results. Left: Compressed images at $q = 5$. Middle: Compression artifact removal results by WCDGAN. Right: Ground truth. WCDGAN successfully removes compression artifacts from highly compressed images while restoring vivid colors.

$u = 0$, $c(u) = \sqrt{\frac{1}{N}}$, otherwise, $c(u) = \sqrt{\frac{2}{N}}$. $c(v)$ has the same function as $c(u)$.

A. RELATED WORK

For a long time, image compression has adopted DCT for block based-transform coding that is successfully applied to various coding standards such as JPEG and MPEG. In addition to the DCT-based image compression, two-dimensional discrete wavelet transform (2D-DWT) decomposes an image into low and high frequency components, i.e. subbands, which can simultaneously capture the spatial and frequency information. The DWT-based image compression performs compression by quantizing and encoding the low and high frequency components. Bose *et al.* [3] used vector quantization (VQ) to compress medical images, and then embedded watermark into the compressed images. They analyzed the effect of watermarking on the content of medical images. In recent years, JPEG compression artifact removal [4]–[7] [8]–[11] [12] and image denoisingl [13]–[15] has been treated as computer vision tasks. Existing methods are roughly classified into two categories: traditional image processing approaches and recent deep learning schemes. The traditional image processing approaches pay more attention to the spatial and frequency domains. The spatial domain methods mainly focus on restoring flat areas [16], some edges [17] and textures [18]. The frequency domain methods mainly utilize DCT coefficients as prior information [4], [19], [20]. Shape adaptive (SA)-DCT, proposed by Foi *et al*. [21], combined spatial information of the images and DCT coefficients to accomplish compression artifact removal. Nowadays, deep convolutional neural networks (CNNs) have obtained continuous success in computer vision and image processing tasks. Inspired by image super-resolution based on CNN (SRCNN) [22], Dong *et al*. [5] proposed an end-to-end network structure for JPEG compression artifact removal, named ARCNN, which added one convolution layer to realize feature enhancement. It is the first compression artifact removal work based on deep learning. Following ARCNN, Zhang *et al*. [6] proposed DnCNN for general image denoising tasks, and used it for compression artifact removal. Chen *et al*. [23] proposed a trainable nonlinear reaction-diffusion model (TNRD) based on the fixed number of gradient descent inference steps. However, TNRD is limited in capturing features of image structure. Galteri *et al*. [24] utilized a generative adversarial network (GAN) [25] to remove compression artifacts. Tai *et al*. [7] proposed a persistent memory network (MemNet) which was stacked by memory blocks consisting of a recursive unit and a gate unit to learn explicit persistent memories. A deep multi-scale CNN (DMCNN), proposed by Zhang *et al*. [8], integrated spatial information and frequency information to make full use of DCT coefficient prior, thus producing higher quality results than the previous work. Liu *et al*. [26] developed a multi-level Wavelet CNN (MWCNN) with U-Net architecture [27] for multiple image restoration tasks: JPEG artifact removal, image denoising and image super-resolution.

Nowadays, generative adversarial networks (GANs) have been also rapidly developed and successfully applied to the computer vision field, such as image super-resolution, image inpainting, and style transfer. GAN is usually composed of a generator and a discriminator. The aim of discriminator is to

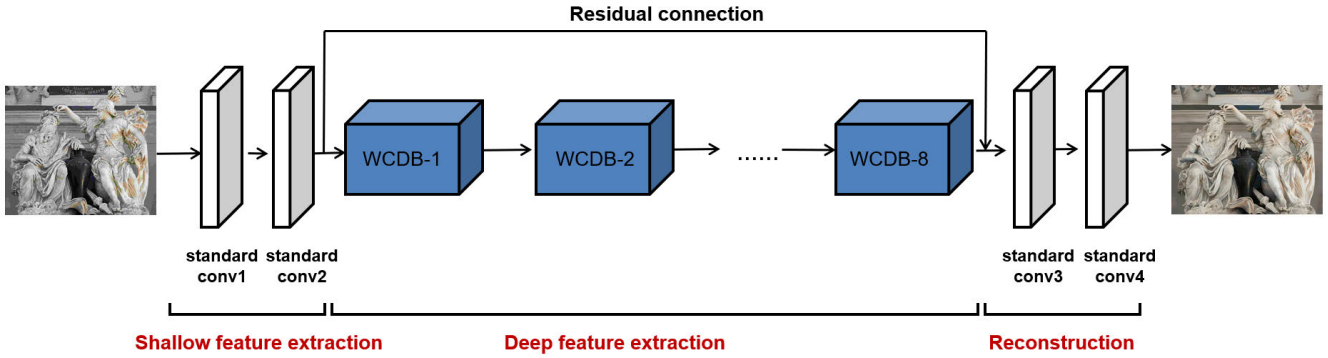


FIGURE 2. Network architecture of the generator in WCDGAN for compression artifact removal. WCDB: Weakly connected dense block. The generator consists of shallow feature extraction, deep feature extraction, and reconstruction. We use 8 WCDBs for deep feature extraction.

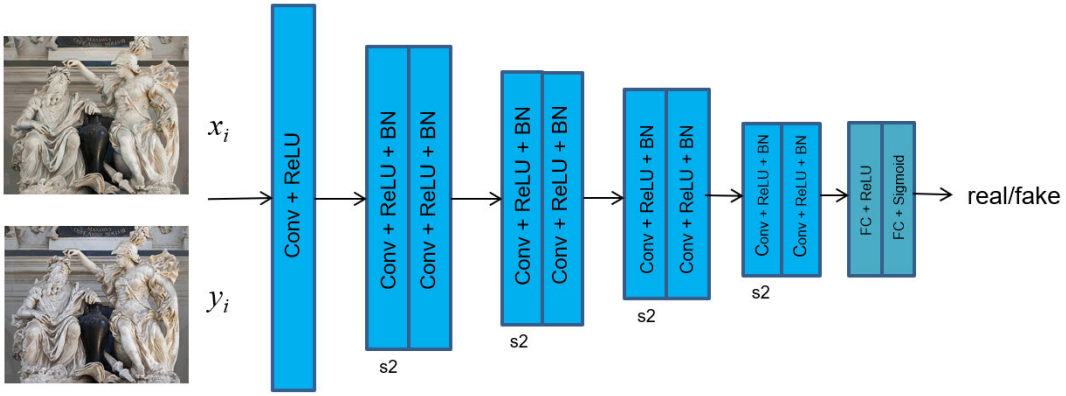


FIGURE 3. Network architecture of the discriminator in WCDGAN. *s*2 means that the stride of convolution is two.

distinguish true and false images. The purpose of generator is to produce more realistic images so that the discriminator cannot distinguish true and false images. The ideal result is to achieve Nash equilibrium where the generator and discriminator can not reduce the cost each other. Despite its great success, GANs still have many defects including generalization and training stability. To solve these problems, many solutions are proposed to improve GANs. Arjovsky *et al*. [28], [29] improved the loss function and minimized the Wasserstein distance between model and data distributions. Then, Mao *et al*. [30] proposed a least squares loss for the discriminator, which made training stable and improve image quality.

B. MOTIVATION

When the compression rate is high, JPEG compression produces noticeable artifacts that seriously degrade image quality as shown in Fig. 1. In the figure, the quality factor q is 5. The most serious distortion is the blocking artifacts that cause banding effect and color distortion in images (see the sky and statues of compressed images). Thus, the compression artifact removal affects color restoration. In this paper, we propose a novel and effective weakly connected dense generative adversarial network for compression artifact removal, called WCDGAN. The proposed WCDGAN aims to remove compression artifacts from highly compressed images when q is lower than 10. We build a weakly connected dense generator for WCDGAN that deploys the attention mechanism [31], [32] and dilated convolution [33], [34] to learn the informative features for compression artifact removal. As shown in Fig. 2, the generator of WCDGAN has three main ingredients: mixed convolution, weakly connected dense block, and mixed attention. First, we combine dilated convolution and standard convolution, i.e. mixed convolution, to enlarge the receptive field and alleviate the grid effect caused by dilated convolution. Second, we provide a weakly connected dense block (WCDB) to reuse the features of the previous convolutional layers in the network and make features more expressive while reducing network parameters and computational cost [35], [36]. Third, we combine channel attention [37] and spatial attention with mixed convolution into WCDB to extract informative features. To consider the color distortion caused by high compression, we implement WCDGAN in the RGB domain, not YCbCr domain. For the discriminator, we use a simple network structure that contains nine convolution layers and two fully connected layers as shown in Fig. 3. Finally, we add a perceptual loss in the loss function to generate realistic images. WCDGAN successfully removes

compression artifacts from highly compressed images while restoring vivid colors (see Fig. 1).

Compared with existing methods, main contributions of this paper are as follows:

- We propose WCDGAN to generate photo-realistic images from highly compressed ones while remove compression artifacts. The generator of WCDGAN has three main ingredients: mixed convolution, weakly connected dense block (WCDB), and mixed attention.
- We combine standard convolution and dilated convolution into mixed convolution to capture a larger receptive field without grid effect. Moreover, we combine channel attention and spatial attention with mixed convolution into WCDB to extract more informative features.
- We add a perceptual loss to generate photo-realistic images with compression artifact removal.

II. PROPOSED METHOD

A. NETWORK ARCHITECTURE

As shown in Fig. 2, the generator of WCDGAN mainly consists of three parts: shallow feature extraction, deep feature extraction and reconstruction. Denote X and Y as the input and output, respectively. Initially, two standard convolutions are used for shallow feature extraction, in which only the second convolutional layer is followed by LeakyReLU activation function. To capture more local information, we use larger convolution kernels in shallow feature extraction, where the sizes of which are 5×5 as follows:

$$F_0 = f_{SF}(X) \tag{5}$$

where $f_{SF}(\cdot)$ denotes the shallow feature extraction operation; and F_0 denotes its output and then is taken as the input to deep feature extraction. The deep feature extraction is a cascade of N WCDB modules (see Fig. 2). As shown in Fig. 4, the WCDB module reuses the features of the previous convolutional layers in the network, thus making features more expressive and reducing network parameters and computational cost. As suggested by ResNet [38] and EDSR [39], we employ a residual connection in this operation to avoid the emergency of gradient vanishing and gradient explosion [40]. Therefore, we get:

$$F_1 = f_{DF}(F_0) + F_0 \tag{6}$$

where $f_{DF}(\cdot)$ denotes the deep feature extraction; and F_1 denotes the output of deep feature extraction and then take it as the input to the reconstruction part, which has the same convolution structure as the shallow feature extraction part. Note that the last convolution layer is not followed by activation function, but the former layer is. In the last two layers, we do not adopt the large convolution kernels, but deploy the small convolution kernels of 3×3. Finally, we obtain the final reconstruction result as follows:

$$Y = f_{REC}(F_1) \tag{7}$$

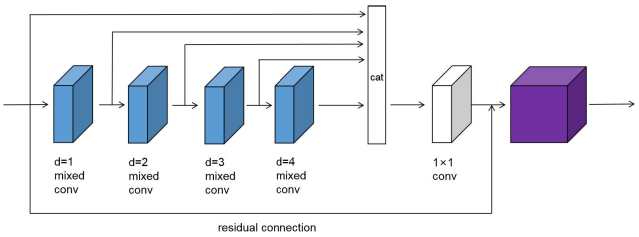


FIGURE 4. Weakly connected dense block (WCDB) module. *cat* means concatenation along channel dimension. The purple block represents the mixed attention module.

Our discriminator is relatively simple to manifest the learning ability of the generator as shown in Fig. 3. This discriminator has nine convolutional layers and two fully connected layers. It adopts a common block composed of Conv, ReLU and Batch Normalization [41]. $s2$ means the stride of convolution is two and the number of convolution kernel of current convolutional layer is doubled. Kernel sizes of all the convolutional layers are set to 3×3.

B. MIXED CONVOLUTION

In low-level computer vision tasks such as image super-resolution, image denoising, and semantic segmentation, the receptive field of the network is an extremely important property [42]. In general, a large receptive field is good for network performance. To sum up, there are the following methods to enlarge the receptive field: (1) make the network deeper; (2) use large convolution kernels; (3) adopt an auto-encoder structure; (4) utilize dilated convolution. The first and second methods inevitably introduce much calculation cost, while the first one can cause the problem of gradient vanishing or gradient explosion. Thus, they are not applicable. The third method deploys a pooling operation in the auto-encoder to reduce the feature size, thus achieving the purpose of expanding the receptive field. However, the pooling operation inevitably causes irreversible loss of information. Excessive pooling operations are unfriendly for pixel-level tasks [43]. Dilated convolution expands the receptive field without increasing parameters and thus is popularly selected. Whereas, it also has its own shortcoming that causes grid effects. To solve this problem, inspired by [44], we design a new convolution module as follows.

This convolution module consists of standard convolution and dilated convolution. As shown in Fig. 5, the input feature acts as inputs for both standard convolution and dilated convolution, then both convolutions are concatenated along the channel dimension with the same weight. Finally, we adopt a 1×1 convolution to compress the feature along the channel dimension and maintain the same shape as the input feature as follows:

$$F_{out} = f_{1\times1}([\delta_{LRe}(f_S(F_{in})), \delta_{LRe}(f_{D=d}(F_{in}))]) \tag{8}$$

where F_{in} and F_{out} represent the input and output of the mixed convolution module, respectively; and $f_S(\cdot)$ denotes the standard convolution, while $f_{D=d}(\cdot)$ denotes the dilated convolution. Note that d is the number of dilation. Additionally,

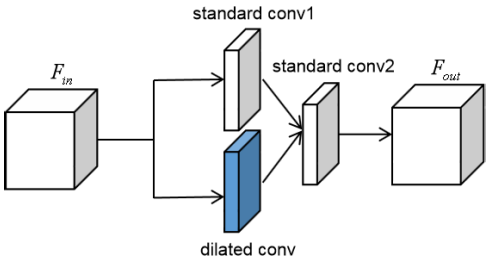


FIGURE 5. Mixed convolution module. Note that the outputs of both standard conv1 and dilated conv are concatenated, which is taken as the input of standard conv2 with kernel size 1 × 1.

$[\cdot]$ represents the concatenation operation and $f_{1\times1}(\cdot)$ is a 1×1 convolution. $\delta_{LRe}(\cdot)$ denotes the LeakyReLU with a negative slope of 0.02.

C. WEAKLY CONNECTED DENSE BLOCK

It has been proved that ResNet solves the gradient vanishing and gradient explosion of a deep neural network to a great extent, which makes the network acquire more powerful representation capabilities. Later, some people have done experiments, in which they randomly dropped some layers of the network and then retrained ResNet. Eventually, they found that the generalization performance of ResNet was significantly improved. It shows that the neural network is not necessarily a hierarchical structure, i.e. a layer in the network can not only rely on the features of the adjacent upper layer, but also on the features of the higher layer. Based on them, Huang *et al*. [45] proposed dense connection to alleviate the problem of gradient vanishing, strengthen feature propagation, and encourage feature reuse. It achieves good performance in many low-level and high-level computer vision tasks.

Nevertheless, with the same deep networks, deploying too many dense blocks brings a lot of burden in the network training, especially in terms of memory consumption, and the performance is not significantly improved. As a result, we remove a part of dense connection and all Batch Normalization layers [41] to achieve satisfactory results while reducing the computational cost. Another difference between WCDGAN and Huang *et al*.'s is that we substitute the proposed mixed convolution for the original standard convolution to obtain a larger receptive field as shown in Fig. 4. The following is a detailed description of this module:

$$X'_{D=1} = f'_{D=1}(X') \tag{9}$$
$$X'_{D=2} = f'_{D=2}(f'_{D=1}(X')) \tag{10}$$
$$X'_{D=3} = f'_{D=4}(f'_{D=2}(f'_{D=1}(X'))) \tag{11}$$
$$X'_{D=4} = f'_{D=8}(f'_{D=4}(f'_{D=2}(f'_{D=1}(X')))) \tag{12}$$

Eqs. (9), (10), (11) and (12) represent the calculation process from the first mixed convolution to the last mixed convolution, where X' denotes input feature, $f'_{D=d}(\cdot)$ means the mixed convolution with d dilations, and $X'_{D=d}$ is the output of the corresponding mixed convolution.

Then, we also utilize a 1×1 convolution to compress the concatenation of both the input feature and output of every mixed convolution. We also add a residual connection to make the training process stable. Next, we calculate:

$$X'' = f_{1\times1}([X', X'_{D=1}, X'_{D=2}, X'_{D=3}, X'_{D=4}]) + X' \tag{13}$$

Finally, the proposed mixed attention module is denoted as $f_{MA}(\cdot)$:

$$X''' = f_{MA}(X'') \tag{14}$$

where X''' is the final output of the deep feature extraction.

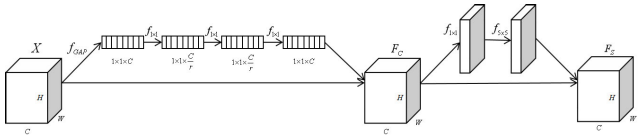


FIGURE 6. Mixed attention module. f_{GAP} denotes global average pooling operation. The number of convolution kernels of the first two 1 × 1 convolutions is $\frac{C}{r}$, and that of the others is C, where r is the downsampling factor. We set r to 4.

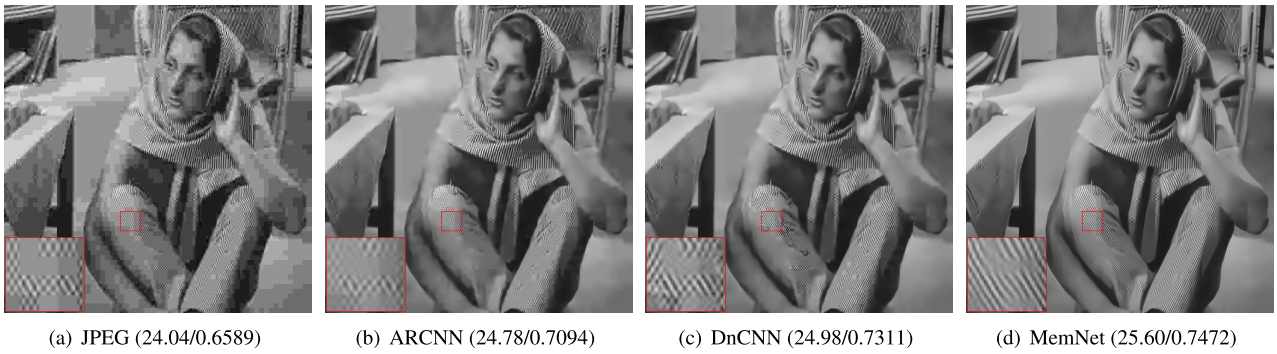
D. MIXED ATTENTION MODULE

Previous methods for compression artifact removal process channel information equally. However, the learned features have different meanings in channel dimension because the features in different channels are learned by different and unrelated convolution kernels. Therefore, to learn more informative features, we exploit channel attention to make the learned features more representational [46], [47]. Therefore, we firstly convert the channel-wise global spatial information into channel weights by using global average pooling [48]. As shown in Fig. 6, $X = [x_1, x_2, \ldots, x_c, \ldots, x_C]$ denotes the input feature with C channels and x_c means c-th channel feature. Let z_c denote the pooling result of the c-th channel feature. Similarly, $Z = [z_1, \ldots, z_c, \ldots, z_C]$. We formulate as follows:

$$z_c = f_{GAP}(x_c) = \frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W} x_c(i,j) \tag{15}$$

where $x_c(i,j)$ is the value at the position (i,j) of the c-th channel feature x_c, $f_{GAP}(\cdot)$ is the global average pooling operation, and the spatial size of feature is $H\times W$.

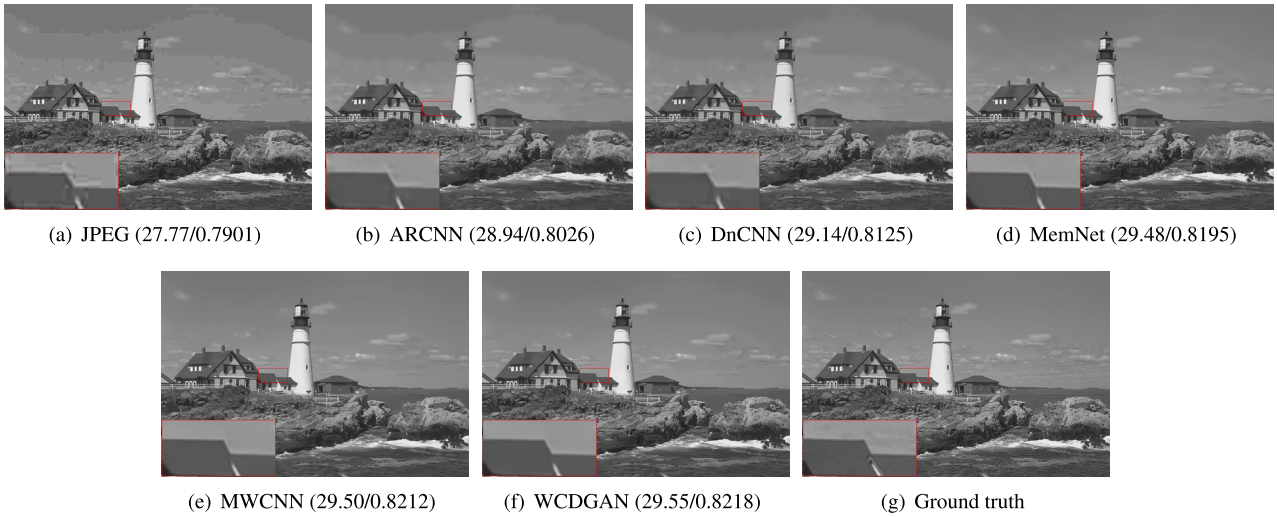
To further exploit channel-wise dependency from the aggregated feature information by the global average pooling, we modify the original structure after that in [49]. We employ a 1×1 convolution followed by LeakyReLU, and then is another 1×1 convolution followed by LeakyReLU. Since we utilize the values less than zero after activation, we use LeakyReLU, instead of ReLU. LeakyReLU makes the weights more robust. At last, it is a 1×1 convolution followed by Sigmoid function. Compared with the original



(a) JPEG (24.04/0.6589) (b) ARCNN (24.78/0.7094) (c) DnCNN (24.98/0.7311) (d) MemNet (25.60/0.7472)

(e) MWCNN (25.71/0.7351) (f) WCDGAN (26.01/0.7494) (g) Ground truth

FIGURE 7. Visual comparison in 'Barbara' (Classic5 dataset) with quality factor 5 (PSNR(dB)/SSIM).



(a) JPEG (27.77/0.7901) (b) ARCNN (28.94/0.8026) (c) DnCNN (29.14/0.8125) (d) MemNet (29.48/0.8195)

(e) MWCNN (29.50/0.8212) (f) WCDGAN (29.55/0.8218) (g) Ground truth

FIGURE 8. Visual comparison in 'Lighthouse' (LIVE1 dataset) with quality factor 10 (PSNR(dB)/SSIM).

structure, we add a middle layer, and LeakyReLU activation function can focus on the weaker elements to a certain extent. Since this channel attention structure fully captures nonlinear interaction between channels and takes the learned relevance as the weight, the features are refined to make the output features have stronger information expression ability as follows:

$$F_C = \sigma(f_{1\times1}(\delta_{LRe}(f^{\frac{C}{r}}_{1\times1}(\delta_{LRe}(f^{\frac{C}{r}}_{1\times1}(Z)))))) \times X \quad (16)$$

where $\delta_{LRe}(\cdot)$ denotes LeakyReLU, $\sigma(\cdot)$ means Sigmoid function, and $f^{\frac{C}{r}}_{1\times1}(\cdot)$ is a 1×1 convolution which acts as channel-downscaling with a ratio r.

Different from the channel attention that emphasizes "what" is meaningful [50], the spatial attention focuses on "where" is an informative part, which is complementary to the channel attention. First, we substitute a 1×1 convolution for the original both average-pooling and max-pooling operations to obtain self-adaptive statistical characteristics. Then, we use a 5×5 convolution to produce a spatial attention map and then refine features as follows:

$$F_S = \sigma(f_{5\times5}(f_{1\times1}(F_C))) \times F_C \quad (17)$$

where $f_{5\times5}(\cdot)$ represents a convolution with the kernel size of 5×5.

E. LOSS FUNCTION

We use L_1 distance to calculate the content loss L_c between the output image I_{out} and the ground truth I_{gt} expressed as follows:

$$L_c = \left\| I_{gt} - I_{out} \right\|_1 \tag{18}$$

Perceptual loss L_p proposed by Johnson *et al*. [51] is to measure perceptual similarity. It is defined as the distance between two features of a pre-trained deep neural network. We first put I_{out} and I_{gt} into VGG-19 to obtain the outputs from a certain convolutional layer, then calculate the L_1 distance between two features as the perceptual loss as follows:

$$L_p = E[\sum_{j=1} \left\| \phi(I_{gt}) - \phi(I_{out}) \right\|_1] \tag{19}$$

where $\phi()$ stands for the last convolution layer of VGG19. L_p is more informative and implements stronger supervision, thus leading to better performance. We use the least squares loss as the adversarial loss in both discriminator and generator. LSGAN [30] not only makes the training process more stable but also generates more gradients than standard GAN, thus improving WCDGAN performance. The discriminator is to distinguish true and fake images and the adversarial loss for the discriminator L_{adv}^D is obtained as follows:

$$L_{adv}^D = E_P[D(I_{gt} - 1)^2] + E_Q[D(I_{out} - 0)^2] \tag{20}$$

where P and Q are the distributions of I_{gt} and I_{out}, respectively. The generator aims to make I_{out} close to I_{gt}, thus the adversarial loss for the generator L_{adv}^G is obtained as follows:

$$L_{adv}^G = E_Q[D(I_{out} - 1)^2] \tag{21}$$

A little noise may have a very big impact on the results. At this time, it is required to add regularization terms for optimization that maintain the smoothness of the image. TV loss L_{tv} is a common regularization term used in conjunction with other losses to constrain noise by reducing the differences between adjacent pixels as follows:

$$L_{tv} = \left\| \nabla_x I + \nabla_y I \right\|_1 \tag{22}$$

where ∇_x and ∇_y calculates the gradient in the x and y directions, respectively. To get L_{tv}, we compute the gradient of each channel in a color image, and then average the sum of them. As the training progresses, total variation in Eq. (22) is minimized and the sensitivity to artifacts is weakened. Moreover, L_{tv} is helpful for smoothing noise generated by GAN. Finally, we combine the losses as the generator loss function for training, which is expressed as follows:

$$L_G = L_c + L_p + L_{adv}^G + L_{tv} \tag{23}$$

III. EXPERIMENTAL RESULTS

A. EXPERIMENTAL SETTINGS

In all experiments, we use 800 training images in DIV2K, released during the NTIRE2017 [52] challenge for image restoration tasks. The LIVE1 [53] dataset, the Classic5 dataset and the validation set of DIV2K are used for evaluation. In terms of the restoration of Y channel, all training and evaluation processes are kept consistent for a fair comparison. Since the methods involved in comparison do not consider color distortion, we design more comparative experiments to verify the feasibility and effectiveness of WCDGAN. We use MATLAB JPEG encoder to generate JPEG-compressed images with $q = 5, 10, 20$. All the approaches including the proposed WCDGAN are implemented with the Pytorch toolbox [54].

We use Adam optimizer with momentum parameters $\beta_1 = 0.5$, $\beta_2 = 0.999$ and initial learning rates 0.0002 and 0.0001 for generator and discriminator, respectively. The learning rate is divided by 2 after every 20000 iterations, while the batch size is set to 16. We extract 80×80 patches with a stride of 75 from image pairs. Adam optimizer is computationally efficient with less memory requirement, which is easy to implement. Thus, Adam optimizer is well suited for problems that the amount of data and parameters are large. It has been reported that Adam optimizer achieves the best performance in optimization [55]. In addition, hyper parameters usually need little adjustment, which makes the use of the optimizer easier and more convenient. That is, Adam optimizer is suitable for a wide range of nonconvex optimization problems in the field of deep learning [56]. Therefore, we choose Adam optimizer to optimize WCDGAN. At the beginning of training, the discriminator has a better ability to distinguish true images from fake ones, thus we only train the generators individually for 4 epochs and then update one step for either generators and discriminators alternately. Unless otherwise specified, convolutional kernel size is 3×3 and the number of convolutional kernel is 64. In the proposed discriminator, when the size of feature map is reduced by 2 times, the number of output channel is enlarged by 2 times.

B. COMPARISON WITH OTHER METHODS ON Y CHANNEL

In YCbCr color space, Y channel represents the luminance information, which mainly contains textures and details. Therefore, we specially get rid of the color influence and directly compare the restoration effect of textures and details. To be a fully convincing comparison, we compare WCDGAN with SA-DCT, ARCNN, TNRD, DnCNN and MemNet. We apply PSNR and SSIM [57] as evaluation metrics for quantitative comparison, which are widely used for image quality assessment. As shown in Table 1, WCDGAN outperforms the others in terms of both PSNR and SSIM. Although the gain of WCDGAN over MWCNN is not very high, the improvement cannot be ignored. WCDGAN reconstructs natural-looking images with clear textures, vivid colors, and good perceptual quality from highly compressed images, i.e. quality factor $q \leq 10$. Since the TV loss in WCDGAN

TABLE 1. Average PSNR (dB) and SSIM values of Y channel on Classic5 and LIVE1. Bold numbers represent the best performance.

Dataset	Quality	JPEG	ARCNN	TNRD	DnCNN	MemNet	MWCNN	WCDGAN
Classic5	5	25.37/0.6708	26.48/0.7047	-/-	26.68/0.7062	26.78/0.7123	27.06/0.7142	**27.29/0.7164**
	10	27.82/0.7800	29.04/0.8111	29.28/0.7992	29.40/0.8026	29.69/0.8107	29.75/0.8112	**29.80/0.8130**
	20	30.12/0.8541	31.16/0.8694	31.47/0.8576	31.63/0.8610	31.90/0.8658	31.91/0.8660	**31.93/0.8661**
LIVE1	5	25.29/0.7081	26.38/0.7214	-/-	26.49/0.7204	26.61/0.7284	26.85/0.7294	**26.93/0.7308**
	10	27.77/0.7905	28.98/0.8217	29.15/0.8111	29.19/0.8123	29.45/0.8193	29.51/0.8215	**29.56/0.8238**
	20	30.07/0.8683	31.29/0.8871	31.46/0.8769	31.59/0.8802	31.83/0.8846	31.86/0.8884	**31.88/0.8896**

TABLE 2. Average PSNR(dB) and SSIM values of RGB images on LIVE1 and validation of DIV2K. Bold numbers represent the best performance.

Dataset	Quality	JPEG	ARCNN	DnCNN	MemNet	MWCNN	WCDGAN
LIVE1	5	24.19/0.7001	24.28/0.7014	24.49/0.7104	24.61/0.7135	24.76/**0.7144**	**26.52**/0.7075
	10	26.77/0.7985	26.98/0.8002	27.19/0.8023	27.41/0.8090	27.45/ **0.8093**	**27.83**/0.7898
Validation Set	5	25.01/0.7491	25.33/0.7501	25.52/0.7512	25.79/0.7541	26.14/**0.7581**	**26.83**/0.7257
	10	28.07/0.8263	28.26/0.8285	28.57/0.8305	28.77/0.8365	29.07/**0.8405**	**29.88**/0.8377

causes smoothing effects, we add L_1 loss and perceptual loss in the loss function to make up for them. Thus, they lead to the detail enhancement of the results. As shown in Figs. 7 and 8, the original blocking and banding artifacts are removed clearly, and the restoration performance of the details and textures in images is also very good. On details, it can be observed from Fig. 7 that WCDGAN restores more continuous texture details but the other methods do not (see the red boxes). In Fig. 8, it is obvious that in the part of the leaves, the proposed method achieves the best restoration performance. Compared with the other methods, our results contain clear and obvious lines, and there are no band-shaped artifacts or block-shaped artifacts. It is obvious that WCDGAN performs the best in details, and the results by the other methods are somewhat rough. From the perspective of overall image quality, our results are better than the others. Therefore, it can be concluded that WCDGAN performs better than the other methods in terms of both local and global visual quality.

C. COMPARISON WITH OTHER METHODS ON RGB IMAGES

The previous experiments mainly focus on the Y channel to show the restoration performance for image textures and details. However, in the high compression rate, the color distortion is also very serious as shown in Fig. 1. Thus, we provide the restoration performance for compressed RGB images. We mainly test the restoration performance of the model on the compressed RGB image with quality factors 5 and 10. Here, we select LIVE1 and validation set of DIV2K as the testing dataset. We provide more deblocking results in Figs. 9-12. From the perspective of the overall visual perception, MWCNN and WCDGAN perform the best in Fig. 9. Both of them remove blocking and banding artifacts better than the others. However, the enlarged areas reveal that our result produces clearer edges and weaker artifacts around the numbers. Additionally, the result of MWCNN also has some slight color distortion compared to WCDGAN. It can observed from Fig. 10 that in the global structure or local details, our results are the best by producing apparent textures and details. Especially, in the enlarged areas, our artifacts are the least among them. A close look at Figs. 11 and 12 reveals that WCDGAN removes most compression artifacts, while successfully restoring textures and details of the images. In the quantitative measurements, it can be observed from Table 2 that perceptual loss and GAN remarkably increase PSNR values while slightly decreasing SSIM values. Thus, perceptual loss and GAN could sacrifice part of the structure information for leading to better perceptual quality in HVS. It focuses on the deep features learned by the model, and emphasizes the semantic information. This is quite different from the image structure in the spatial domain. In a word, GAN structure and perceptual loss are helpful for improving the perceptual quality of human eyes. On the whole, WCDGAN achieves good perceptual quality in the results after compression artifact removal.

D. COMPUTATIONAL COMPLEXITY

As shown in Table 3, we provide comparison of computational complexity in terms of model size and runtime. We compare WCDGAN with state-of-the art methods: ARCNN, DNCNN, MemNet, and MWCNN. For tests, we use a PC with Nvidia GeForce GTX 1080Ti GPU and Intel i7-7700 3.60GHz CPU. It is obvious that the model size of WCDGAN are less than MemNet and MWCNN, while the runtime of WCDGAN is faster than them. Although the model size and runtime of WCDGAN are not the best, WCDGAN keeps a balance between compression artifact removal and computational complexity. The results indicate that WCDGAN is effective for removing compression artifacts from highly compressed images with low complexity.

E. ABLATION STUDY

We examine the impact of various important modules on performance. First, we evaluate the network performance with and without the mixed convolution. No mixed convolution

(a) JPEG (24.18/0.7000) (b) ARCNN (24.25/0.7011) (c) DnCNN (24.45/0.7100) (d) MemNet (24.62/0.7132)

(e) MWCNN (24.75/0.7145) (f) WCDGAN (26.50/0.7076) (g) Ground truth

FIGURE 9. Visual comparison in 'Sailing' (LIVE1 dataset) with quality factor 5 (PSNR(dB)/SSIM).

TABLE 3. Comparison of computational complexity in terms of model size and runtime. For tests, we use a PC with Nvidia GeForce GTX 1080Ti GPU and Intel i7-7700 3.60GHz CPU.

	ARCNN	DnCNN	MemNet	MWCNN	WCDGAN
Model size (M)	0.1	0.6	2.9	25.1	1.7
Runtime (s)	0.5722	0.8071	1.6937	1.4814	1.4606

TABLE 4. Ablation study on mixed convolution and mixed attention in Classic5 and LIVE1 datasets.

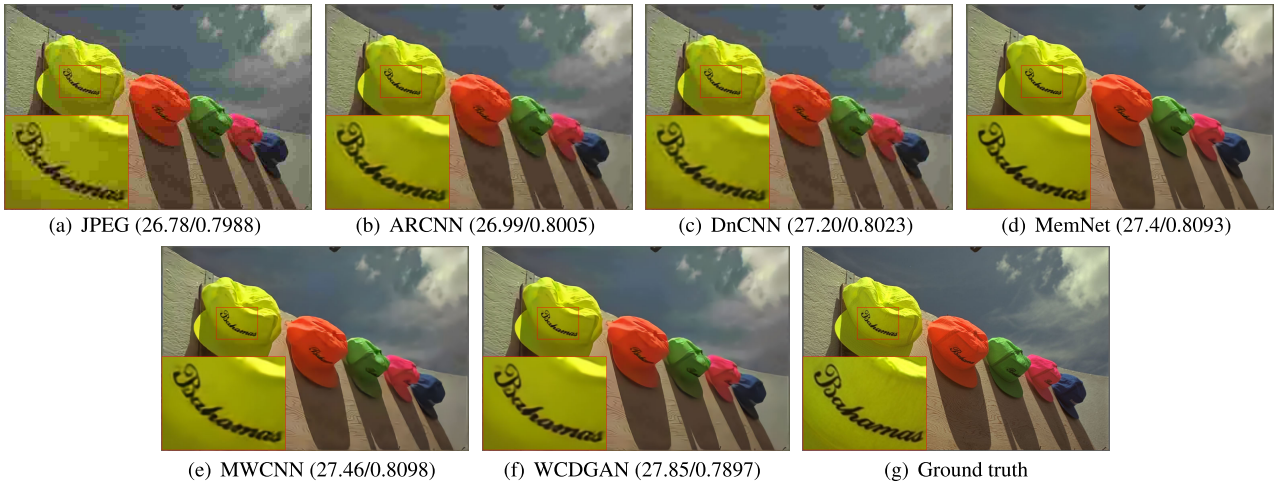
Dataset	q	Proposed	Without Mixed Convolution	Without Mixed Attention
Classic5	5	**26.88/0.7164**	26.69/0.6981	26.79/0.7023
	10	**29.80/0.8291**	29.62/0.8110	29.73/0.8148
	20	**31.93/0.8833**	31.72/0.8652	31.85/0.8691
LIVE1	5	**26.86/0.7308**	26.72/0.7158	26.79/0.7168
	10	**29.56/0.8432**	29.42/0.8266	29.48/0.8290
	20	**31.88/0.9072**	31.73/0.8911	31.79/0.8931

means that we replace the mixed convolution with the standard convolution. As shown in Table 4, when the mixed convolution module is not deployed, in Classic5 dataset the PSNR value is reduced by about 0.2dB and the SSIM value is reduced by about 0.018. In LIVE1 dataset, the PSNR value is reduced by about 0.15dB and the SSIM value is reduced by about 0.016. They are average performance over three different quality factors. It is enough to show that the mixed convolution plays an important role in improving the performance of WCDGAN. The results indicate that the large receptive field can improve the network performance. Also, we explore the effects of the attention mechanism. As shown in the figure, the PSNR is reduced by about 0.08dB and the SSIM is reduced by about 0.014 in Classic5 dataset, while the PSNR is reduced by about 0.09dB and the SSIM is reduced by about 0.014 in LIVE1 dataset. It can be figured out that the mixed attention mechanism has a positive impact on the

(a) JPEG (25.00/0.7490) (b) ARCNN (25.30/0.7502) (c) DnCNN (25.53/0.7511) (d) MemNet (25.80/0.7544)

(e) MWCNN (26.13/0.7579) (f) WCDGAN (26.82/0.7258) (g) Ground truth

FIGURE 10. Visual comparison in 'Owl' (validation dataset of DIV2K) with quality factor 5 (PSNR(dB)/SSIM).

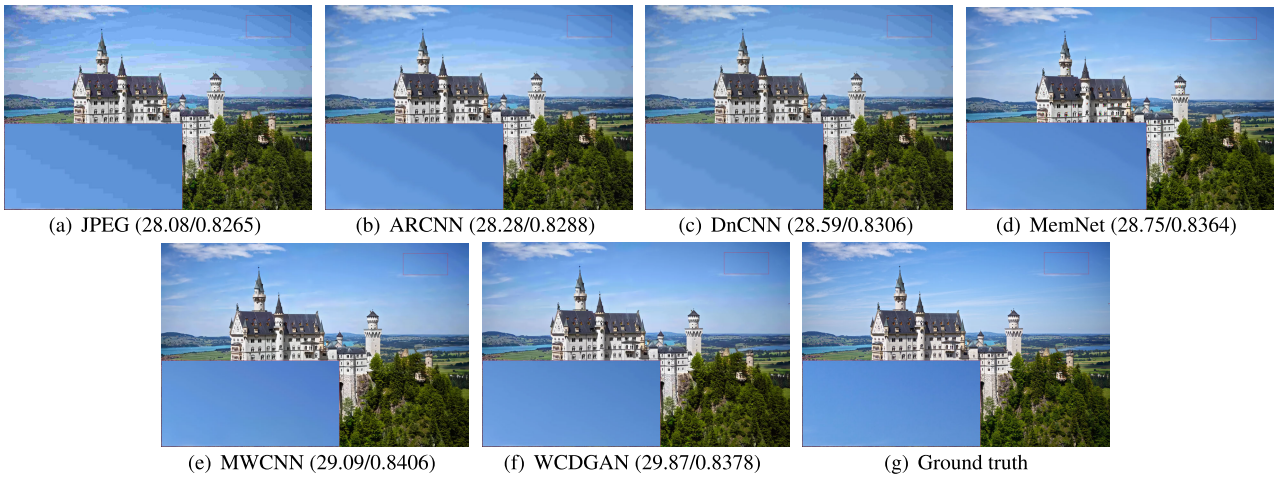


(a) JPEG (26.78/0.7988) (b) ARCNN (26.99/0.8005) (c) DnCNN (27.20/0.8023) (d) MemNet (27.4/0.8093)

(e) MWCNN (27.46/0.8098) (f) WCDGAN (27.85/0.7897) (g) Ground truth

FIGURE 11. Visual comparison in 'Caps' (LIVE1 dataset) with quality factor 10 (PSNR(dB)/SSIM).

compression artifact removal. The experiments demonstrate that both mixed convolution and mixed attention are effective and valuable for compression artifact removal. Furthermore, we perform another ablation study on perceptual loss and GAN. Table 5 indicates that perceptual loss and GAN remarkably increase PSNR values while slightly decreasing SSIM



(a) JPEG (28.08/0.8265) (b) ARCNN (28.28/0.8288) (c) DnCNN (28.59/0.8306) (d) MemNet (28.75/0.8364)

(e) MWCNN (29.09/0.8406) (f) WCDGAN (29.87/0.8378) (g) Ground truth

FIGURE 12. Visual comparison in 'Castle' (validation dataset of DIV2K) with quality factor 10 (PSNR(dB)/SSIM).

(a) JPEG (b) Without perceptual loss and GAN (c) With perceptual loss and GAN (d) Ground truth

FIGURE 13. Ablation study on perceptual loss and GAN in 'Owl' (validation dataset of DIV2K) with quality factor 10.

TABLE 5. Ablation study on perceptual loss and GAN in LIVE1 and DIV2K validation datasets in terms of PSNR and SSIM. The numbers represent PSNR(dB)/SSIM, while the bold ones represent the best performance.

Dataset	q	Proposed	Without perceptual loss and GAN
LIVE1	5	**26.52**/0.7075	24.88/**0.7151**
	10	**27.83**/0.7898	27.48/**0.8097**
DIV2K validation	5	**26.83**/0.7257	26.17/**0.7604**
	10	**29.88**/0.8377	29.13/**0.8430**

TABLE 6. Ablation study on perceptual loss and GAN in LIVE1 and DIV2K validation datasets in terms of Inception Score (IS) [58] or Fréchet Inception Distance (FID) [59]. The numbers represent IS/FID values, while the bold ones represent the best performance.

Dataset	q	Proposed	Without perceptual loss and GAN
LIVE1	5	**2.9788/45.8891**	2.8395/56.8587
	10	**2.9895/42.5418**	2.8556/53.8488
DIV2K validation	5	**2.9856/44.2658**	2.8569/54.7456
	10	**2.9947/41.5369**	2.8659/51.9858

values. That is, perceptual loss and GAN sacrifice part of the structure information benefiting to better visual perception in HVS. In addition, we provide Inception Score (IS) [58] and Fréchet Inception Distance (FID) [59] in Table 6 to evaluate the realism of the generated images. As shown in the table, the perceptual loss and GAN contribute to the generation of photo-realistic images. Note that bigger IS and smaller FID indicate better performance. We provide an ablation study on the perceptual loss and GAN in Fig. 13. It can be observed that the result without perceptual loss and GAN contains serious artifacts, which seriously degrades the visual quality. However, the perceptual loss and GAN almost completely remove the artifacts while generating clear and natural textures in the result, thus greatly improving the visual quality.

IV. CONCLUSION

In this paper, we have proposed a novel WCDGAN for artifact removal of highly compressed images. We have attempted to remove compression artifacts and recover high-quality images from highly compressed images. The mixed convolution is able to enlarge the receptive field of the network mitigating the grid effect of dilated convolution. WCDB enhances feature reuse and makes features more expressive. Moreover, the mixed attention module makes WCDGAN learn informative features. Experimental results demonstrate that WCDGAN reconstructs natural-looking images with clear textures, vivid colors, and good perceptual quality from highly compressed images, even in $q = 5$. WCDGAN outperforms the state-of-the-art models for compression artifact removal in terms of both visual quality and quantitative measurements.

Although WCDGAN has achieved good performance in compression artifact removal, it has a limit of implementing a lightweight model for practical applications. Therefore, we will investigate yielding a lightweight network for compression artifact removal by replacing common convolution with depthwise separable convolution.

REFERENCES

[1] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consum. Electron.*, vol. 38, no. 1, pp. 30–44, Feb. 1992.
[2] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-100, no. 1, pp. 90–93, Jan. 1974.
[3] S. Bose, Madhulika, S. Acharjee, S. R. Chowdhury, S. Chakraborty, and N. Dey, "Effect of watermarking in vector quantization based image compression," in *Proc. Int. Conf. Control, Instrum., Commun. Comput. Technol. (ICCICCT)*, Jul. 2014, pp. 503–508.
[4] T. Chen, H. R. Wu, and B. Qiu, "Adaptive postfiltering of transform coefficients for the reduction of blocking artifacts," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 5, pp. 594–602, May 2001.
[5] C. Dong, Y. Deng, C. C. Loy, and X. Tang, "Compression artifacts reduction by a deep convolutional network," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 576–584.
[6] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.
[7] Y. Tai, J. Yang, X. Liu, and C. Xu, "MemNet: A persistent memory network for image restoration," in *Proc. IEEE Conf. Comput. Vis.*, Oct. 2017, pp. 4539–4547.
[8] X. Zhang, W. Yang, Y. Hu, and J. Liu, "DMCNN: Dual-domain multi-scale convolutional neural network for compression artifacts removal," in *Proc. IEEE Conf. Image Process.*, Oct. 2018, pp. 390–394.
[9] 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. 20160–20170, 2020.
[10] J. Li, Y. Wang, H. Xie, and K.-K. Ma, "Learning a single model with a wide range of quality factors for JPEG image artifacts removal," *IEEE Trans. Image Process.*, vol. 29, pp. 8842–8854, 2020.
[11] J. Liu, D. Liu, W. Yang, S. Xia, X. Zhang, and Y. Dai, "A comprehensive benchmark for single image compression artifact reduction," *IEEE Trans. Image Process.*, vol. 29, pp. 7845–7860, 2020.
[12] 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 Trans. Circuits Syst. Video Technol.*, vol. 30, no. 4, pp. 1121–1135, Apr. 2020.
[13] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2862–2869.
[14] D.-G. Kim, M. Hussain, M. Adnan, M. A. Farooq, Z. H. Shamsi, and A. Mushtaq, "Mixed noise removal using adaptive median based non-local rank minimization," *IEEE Access*, vol. 9, pp. 6438–6452, 2021.
[15] Z. H. Shamsi, D.-G. Kim, M. Hussain, and R. M. B. K. Sajawal, "Low-rank estimation for image denoising using fractional-order gradient-based similarity measure," *Circuits, Syst., Signal Process.*, vol. 40, no. 10, pp. 4946–4968, Oct. 2021.
[16] P. List, A. Joch, J. Lainema, G. Bjontegaard, and M. Karczewicz, "Adaptive deblocking filter," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 614–619, Jul. 2003.
[17] H. C. Reeve and J. S. Lim, "Reduction of blocking effects in image coding," *Opt. Eng.*, vol. 23, no. 1, 1984, Art. no. 230134.
[18] C. Wang, J. Zhou, and S. Liu, "Adaptive non-local means filter for image deblocking," *Signal Process., Image Commun.*, vol. 28, no. 5, pp. 522–530, 2013.
[19] A. Foi, V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive DCT as an overcomplete denoising tool," in *Proc. Int. TICSP Workshop Spectral Methods Multirate Signal Process.*, 2005, pp. 164–170.
[20] A. Foi, K. Dabov, V. Katkovnik, and K. Egiazarian, "Shape-adaptive DCT for denoising and image reconstruction," *Proc. SPIE*, vol. 6064, pp. 1–18, Feb. 2006.
[21] A. Foi, V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images," *IEEE Trans. Image Process.*, vol. 16, no. 5, pp. 1395–1411, May 2007.
[22] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. Eur. Conf. Comput. Vis.* Zürich, Switzerland: Springer, 2014, pp. 184–199.
[23] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1256–1272, Jun. 2016.
[24] L. Galteri, L. Seidenari, M. Bertini, and A. D. Bimbo, "Deep generative adversarial compression artifact removal," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4826–4835.
[25] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[26] P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo, "Multi-level wavelet-CNN for image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 773–782.
[27] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput.-Assist. Intervent.* Munich, Germany: Springer, 2015, pp. 234–241.
[28] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*.
[29] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of Wasserstein GANs," 2017, *arXiv:1704.00028*.
[30] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," 2016, *arXiv:1611.04076*.
[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[32] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," 2018, *arXiv:1805.08318*.
[33] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, *arXiv:1511.07122*.
[34] G. Seif and D. Androutsos, "Large receptive field networks for high-scale image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 763–772.
[35] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.
[36] T. Tong, G. Li, X. Liu, and Q. Gao, "Image super-resolution using dense skip connections," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 4799–4807.
[37] Y. Hu, J. Li, Y. Huang, and X. Gao, "Channel-wise and spatial feature modulation network for single image super-resolution," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 11, pp. 3911–3927, Nov. 2020.
[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[39] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 136–144.
[40] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.

[41] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*.
[42] R. Wang, M. Gong, and D. Tao, "Receptive field size versus model depth for single image super-resolution," *IEEE Trans. Image Process.*, vol. 29, pp. 1669–1682, 2019.
[43] F. Saeedan, N. Weber, M. Goesele, and S. Roth, "Detail-preserving pooling in deep networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9108–9116.
[44] C. Liu, Z. Shang, and A. Qin, "A multiscale image denoising algorithm based on dilated residual convolution network," in *Proc. Chin. Conf. Image Graph. Technol.* Beijing, China: Springer, 2019, pp. 193–203.
[45] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," 2016, *arXiv:1608.06993*.
[46] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.-S. Chua, "SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5659–5667.
[47] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3156–3164.
[48] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 286–301.
[49] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.
[50] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 3–19.
[51] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," 2016, *arXiv:1603.08155*.
[52] E. Agustsson and R. Timofte, "NTIRE 2017 challenge on single image super-resolution: Dataset and study," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 126–135.
[53] H. Sheikh, Z. Wang, L. Cormack, and A. C. Bovik. (2005). *Live Image Quality Assessment Database Release 2*. [Online]. Available: http://live.ece.utexas.edu/research/quality
[54] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *Proc. Adv. Neural Inf. Process. Syst. Workshop*, Long Beach, CA, USA, 2017.
[55] M. N. Y. Ali, M. G. Sarowar, M. L. Rahman, J. Chaki, N. Dey, and J. M. R. S. Tavares, "Adam deep learning with SOM for human sentiment classification," *Int. J. Ambient Comput. Intell.*, vol. 10, no. 3, pp. 92–116, Jul. 2019.
[56] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[57] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[58] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.
[59] 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," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6626–6637.

BINGHUA XIE received the B.S. degree in communication engineering from Xidian University, China, in 2018, where he is currently pursuing the M.S. degree in electronic engineering. His main research interests include image processing, computer vision, and deep learning.

HAO ZHANG received the B.S. degree in electronic information engineering from Changan University, China, in 2020. He is currently pursuing the M.S. degree in electronic engineering with Xidian University, China. His research interests include image processing, video coding, and deep learning.

CHEOLKON JUNG (Member, IEEE) is a Born Again Christian. He received the B.S., M.S., and Ph.D. degrees in electronic engineering from Sungkyunkwan University, Republic of Korea, in 1995, 1997, and 2002, respectively. He was a Research Staff Member with the Samsung Advanced Institute of Technology, Samsung Electronics, Republic of Korea, from 2002 to 2007. He was also a Research Professor with the School of Information and Communication Engineering, Sungkyunkwan University, from 2007 to 2009. Since 2009, he has been with the School of Electronic Engineering, Xidian University, China, where he is currently a Full Professor and the Director of the Xidian Media Laboratory. His main research interests include image and video processing, computer vision, pattern recognition, machine learning, computational photography, video coding, virtual reality, information fusion, multimedia content analysis and management, and 3DTV.