The Concatenated Dynamic Convolutional and Sparse Coding on Image Artifacts Reduction

Linna Yang[1][0000−0001−7152−2760] and Ronny Velastegui[1][0000−0001−8628−9930]

[1]Norwegian University of Science and Technology, Gjøvik, Norway.
linnay@stud.ntnu.no

Abstract. In order to enhance compressed JPEG image, a deep convolutional sparse coding network is proposed in this article. The network integrates state-of-the-art dynamic convolution to extract multi-scale image features, and uses convolutional sparse coding to separate image artifacts to generate coded feature for the final image reconstruction. Since this architecture consolidates model-based convolutional sparse coding with deep neural network, that allow this method has more interpretability. Also, compared with the existing network, which uses a dilated convolution as a feature extraction approach, this proposed concatenated dynamic method has improved de-blocking result in both numerical experiments and visual effect. Besides, in the higher compressed quality task, the proposed model has more pronounced improvement in reconstructed image quality evaluations.

Keywords: Deep learning · Image reconstruction · Sparse coding · Dynamic convolution

1 Introduction

Generally, there are two main image compression methods used recently. One of them is lossless compression, which usually exploit statistical redundancy in such a way as to represent the sender's data more concisely, but nevertheless perfectly [18]. For example, the run-length encoding [17], is often used in medical, high-tech and comics fields. And the variable-length coding (VLC) [29], whose the most significant property is more frequent symbols receive shorter codes. Several famous algorithms like Shannon-Fano Algorithm [32], Huffman coding [23] are all VLC.

Another one is lossy compression, this compression algorithm does not deliver high enough compression ratios and are widely used in the web and other areas that do not mind the loss of fidelity but require drastically reduction of bit rate [34]. Hence, most multimedia compression algorithms are lossy and commonly apply concept of perceptually lossless compression where the perceptual distortion metrics are needed. Although lossy compression methods can give acceptable result in most cases, there is always a trade-off between the compressed rate and distortion. They will still introduce compression artifacts. These artifacts and blocking might severely reduce the visually perceived quality and subsequent computer vision systems [31].

JPEG is one of the most commonly used lossy image compression standard, primarily used for natural images, was developed by the Joint Photographic Experts Group and was formally accepted as an international standard in 1992 [21]. JPEG compression is achieved by implementing discrete cosine transform (DCT) [1], a type of Fourier-related transform, that also used in more recent high-efficiency image file format (HEIF) [15]. The role of DCT is to decompose the original signal into its constant magnitude and periodic variations components. And then, the inverse discrete cosine transform (IDCT) is used to reconstruct the signal. The effectiveness of the DCT transforms coding method in JPEG relies on three major observations: Spatial redundancy; Lower sensitivity to loss in higher spatial frequencies in human eyes; Less visual acuity for color than gray [22].

There are several steps in JPEG Image Compression. After color conversion and chroma sampling, DCT is implied on image blocks. In this step, each image is divided into 8×8 block. Due to the DCT has a strong "energy concentration" characteristic: most of the natural signal (including sound and image) energy is concentrated in the additional part after the discrete cosine transform. It can compress the size of pictures to a pretty small level [22]. By using block, however, it has the effect of isolating each block from its neighboring context, sometimes it has visible change from block to block, that is why JPEG images look choppy ("blocky") when the user specifies a high compression ratio [5]. This discontinuity at the boundaries of block could sharply degrade the visual perception of image. Because of that, artifacts reduction of lossy compression is a necessary task in computer vision.

This article describes a JPEG image reconstruction method that combines the advantages of two popular image enhancement types in deep learning and utilizes state-of-the-art dynamic convolution. Compared with the previous counterpart with dilation convolution, the complement of dynamic convolution of this proposed model achieves better restoration performance in both numerical result and visual perception. Furthermore, with the improvement of picture quality, the role of new loss function becomes more notable and positively impacts the deblocking performance.

2 Background

In order to enhance the compressed images quality, various strategies have been proposed. Generally, these image reconstruction methods can be roughly divided into two types. One is model-based [10, 25–27], and the other one is learning-based [4, 12, 18, 36, 39]. The former is usually modeled using domain knowledge, especially some specific physical meaning, but the cost is time-consuming optimization. The latter focus on learn non-linear mapping function from training dataset directly, that allows this kind of methods have faster speed, while have less sufficient interpretation. These two classes of strategy have complementary advantages, although learning-based methods usually have more agreeable performance compared with another one.

Early JPEG image reconstruction methods depended on design heavily. For example, the filtering in the image domain or the transform domain, like using joint image domain filtering method [14]. Another way is regarding the artifacts reduction as the ill-posed inverse problem through optimization, like non-local self-similarity property. In this direction, sparsity as an effective technique to solve this ill-posed problem has been fully explored.

Moreover, deep convolutional neural networks (CNNs) made substantial progress in the past decade. In the image compressed and artifacts reduction task, this learning-based model has an agreeable ability in learning a nonlinear mapping from image pairs, which consist of the uncompressed original image and its compressed JPEG counterpart. [18] first proposed the deep convolutional neural networks method for JPEG artifacts reduction by utilizing the super-resolution network [8] as its cornerstone. Because of the inspiration from dense connection and residual learning, some deep CNNs approaches were introduced for image reconstruction and denoising [32, 38].

Recent work gives some combinable solutions to keep the merits of both model-based and learning-based techniques, using a competitive deep learning model with model-based sparse coding (SC) [35]. Furthermore, Xueyang Fu et al. [11] adopt dilated convolutions to address the different image qualities problem. They propose a method that implements convolutional sparse coding (CSC) on fully connected layers. Due to the space-invariant characteristics of CSC, it can process the entire image, typically suitable for some vision tasks without high-level requests. This method is based on the work of Xueyang Fu et al., but using a different way to solve the multi-scale feature extraction issue and made some improvements in the loss function.

This proposed deep learning model has better explainability that benefits from the idea of utilization of CSC [16], which learned iterative shrinkage threshold algorithm (LISTA) can be applied to separate artifacts. And since the deep CSC part is built from the idea of the classical optimization algorithm,that making this model more structured and clear. Additionally, dynamic convolution makes it possible to process multiple image qualities. This high flexibility allows the proposed method to have a broader range of applications.

3 Methodology

This proposed network contains three main integrants designed for each specific task: multiscale feature extraction (using dynamic convolution), followed by a classic CSC approach – LISTA, and the final image reconstruction. Figure 1 shows the workflow of this model, which is quite composable, which using compressed JPEG image $\mathbf{J}$ as input and achieved de-blocking image $\mathbf{O}$ as output. In this networks, firstly, dynamic convolutions are designed for compressed image $\mathbf{J}$ feature extraction in three different clarities. Secondly, a convolutional LISTA is used to build sparse code $\mathbf{X}$ for artifacts identification and separation. Thirdly, to predict the final reconstructed image $\mathbf{O}$, the trained sparse codes $\mathbf{U_R}$ go to generate the residual $\mathbf{H}$.

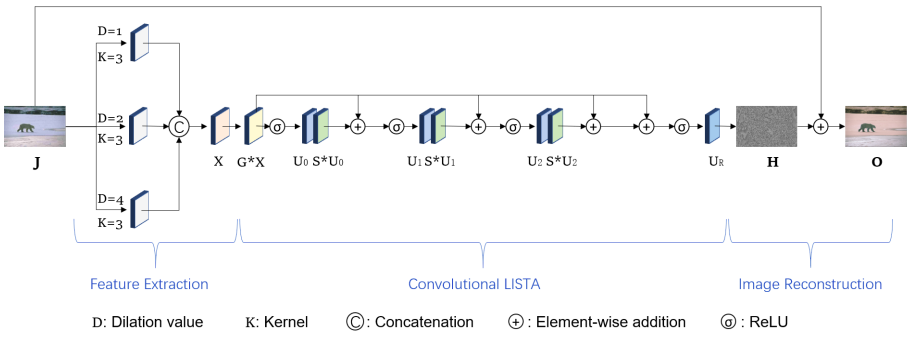


Fig. 1: The workflow of proposed network for image artifacts reduction

3.1 Network architecture

Dynamic Convolution Due to the compression quality of JPEG could be varied, only using one fixed model cannot meet the demand. In order to solve this problem, some learning-based strategies build several models based on different image qualities [8], some of them at the cost of greater parameter burden, train networks for the increased receiving field [38], some of them apply dilated convolutions instead [11]. In this network, dynamic convolution is adopted to handle the multi-scale feature extraction procedure.

The propose of dynamic convolution is mainly from the need for light-weight CNNs, typically for mobile devices that require to enable many functions and generate output in real-time. And in CNN architecture, it usually faces the problem that when computational constraint goes lower, the performance shows obviously degradation, since the extremely low computational constraint not only influences the depth of network but also the number of channels, which are pivotal for network performance [3].

Chen et al. [3] come up with a new operator design that presents aggregated multiple convolution kernels with constant network both in depth and width, which has remarkably improvement compared with its single kernel counterpart. The goal of this dynamic convolutional neural networks (DY-CNNs) is to figure out the dilemma between network efficiency and acceptable computational cost. DY-CNN does not increase either the layers or the channels of network, but the introduced K kernels, which determined by different input, play an essential role in enhancing the model capability.

Firstly, this DY-CNN aggregates K linear functions, the weight and bias are defined as equation (1), where the π_k is between 0 and 1, representing the attention weight of various linear function, that differ from every input images. Thus, the dynamic perception model is defined in equation (2), a non-linear function with a higher representation ability.

$$\tilde{\boldsymbol{W}}(x) = \sum_{k=1}^{K} \pi_k(x)\tilde{\boldsymbol{W}}_k, \tilde{\boldsymbol{b}}(x) = \sum_{k=1}^{K} \pi_k(x)\tilde{\boldsymbol{b}}_k \tag{1}$$

$$\boldsymbol{y} = g(\tilde{\boldsymbol{W}}^T(x)x + \tilde{\boldsymbol{b}}(x)) \tag{2}$$

In addition, assembled parallel convolution kernels share the same input and output channels. Hence the network width or depth is not changed, which gives dynamic convolution great compatibility. And although the introduction of attention weights and K kernels brings some extra computational cost, this induced cost is still negligible compared with the convolution operation.

In this networks, a dynamic convolution using classic CNN design [3] (figure 1) is used in the first step. Before the beginning, by using the squeeze-and-excitation method [19], the kernel attentions can be computed. At first, the global average pooling layers squeezes the spatial information. It further goes through two fully connected (FC) layers with one ReLU between them and one softmax after them to create normalized attention weights. After the attention is all computed, the aggregated convolution output is passed a batch normalization (BN) layer and goes to the last activation in this part.

Specifically, to get a more adaptable solution of this model, using three different dilation values (In this work, 1,2,4 are used), a concatenated feature extraction layer is generated. In equation (3), $\boldsymbol{X}_D$ is the output of the dynamic convolutions using different dilation values, *concat* indicates the concatenation.

$$\mathbf{X} = concat(\mathbf{X}_D) \tag{3}$$

Then, this concatenated $\boldsymbol{X}$ after extracted image features goes into the following procedure.

Convolutional LISTA In order to obtain the complementary advantages of learning-based and model-based methods, the middle part of this network is designed to use LISTA [38], which is a classic CSC method.

Sparse coding is to use a set of over-complete bases to represent a vector, and the obtained vector has a certain sparseness. At the same time, the input vector is a linear combination of these bases. And the issue it originally handled is to find a suitable sparse code that can minimize function 4 with L1 Regularization. In this function, $\boldsymbol{x}$ is the input, and the $\boldsymbol{u}$ is the sparse code correspondingly, with Φ as an over-complete dictionary. And λ is positive a parameter, F represents Frobenius norm.

$$\arg\min_{\boldsymbol{u}} \|\mathbf{x} - \mathbf{\Phi u}\|_F + \lambda\|\mathbf{u}\|_1 \tag{4}$$

To find a way to solve the problem in function 4, the following iterative equation (5) that to obtain optimized result was introduced in iterative shrinkage threshold algorithm (ISTA) originally, where the σ_θ is the shrinkage function that with a θ as threshold and r is the iteration parameter. L is a constant and must

be the upper limit of the maximum eigenvalue of $\mathbf{\Phi}^T\mathbf{\Phi}$. In addition, note that $\mathbf{G}$ and $\mathbf{S}$ have a coupling relationship, that will cause the degradation of flexibility and capacity in the model. Using independent kernels to $\mathbf{G}$ and $\mathbf{S}$ respectively helps taking full advantage of deep learning. LISTA learn parameters from data, which approximate the SC of ISTA, that can give it faster speed, especially in real-time implements.

$$\begin{aligned}\mathbf{u}_r &= \sigma_\theta(\mathbf{u}_{r-1} + \frac{1}{L}\mathbf{\Phi}^T(\mathbf{x} - \mathbf{\Phi}\mathbf{u}_{r-1})) \\ &= \sigma_\theta(\frac{1}{L}\mathbf{\Phi}^T\mathbf{x} + (\boldsymbol{I} - \frac{1}{L}\mathbf{\Phi}^T\boldsymbol{\Phi})\mathbf{u}_{r-1}) \\ &= \sigma_\theta(\mathbf{G}\mathbf{x} + \mathbf{S}\mathbf{u}_{r-1}) \\ \mathbf{u}_0 &= \sigma_\theta(\mathbf{G}\mathbf{x})\end{aligned} \tag{5}$$

Although CSC has been applied to several image reconstruction tasks, the multiple features extracted through these methods are actually shifted versions of the same one. To solve this problem, CSC methods have been introduced to construct the objective function in a shift-invariant manner, which can be represented in the following ways:

$$\arg\min_{\mathbf{w},\mathbf{U}} \|\mathbf{X} - \sum_{m=1}^{M} \mathbf{w}(m) * \mathbf{U}(m)\|_F + \lambda \sum_{m=1}^{M} \|\mathbf{U}(m)\|_1 \tag{6}$$

The approximated result of input image $\mathbf{X}$ can be achieved when $\mathbf{w} * \mathbf{U}$, where $\mathbf{w}$ is convolutional dictionaries and U is sparse coefficient and the number of both of them are M. By converting the kernel into a circulant matrix, the convolution operation can be executed as a matrix multiplication [28]. Because of that, the convolutional sparse coding model could be regarded as a special case of the normal sparse coding model. Thus, compared with function 4, function 6 uses convolutional operation instead to address this optimization issue.

After this, the feature map $\mathbf{U}_r$ can obtained from sparse feature coefficients by embedding the convolutional dictionaries to kernel $\mathbf{G}$ and $\mathbf{S}$, which are learnt from last stage. Rectified Linear Unit (ReLU) [24] is used as the non-linear activation function, because sparsity can be introduced by ReLU.

Image restoration At this time, the final feature maps $\mathbf{U}_r$ are achieved by the R iterations in the last step, and can be used in generating output image. In this method, the residual image $\mathbf{H}$ mapped by $\mathbf{U}_r$ (7), that formed residuals are used to simplify learning problem [38]. $\mathbf{W}_R$ and $\mathbf{b}_R$ are the convolutional weights and bias parameters respectively.

$$\mathbf{H} = \mathbf{W}_R * \mathbf{U}_R + \mathbf{b}_R \tag{7}$$

Because of preservation of estimated residual in $\mathbf{H}$, the output image $\mathbf{O}$ which removed blocking artifacts can be calculated as following (8):

$$\mathbf{O} = \mathbf{J} + \mathbf{H} \tag{8}$$

3.2 Loss function

Although mean squared error (MSE) is the widely used in image reconstruction tasks, it usually produces results that are too smooth because the square penalty does not work well at the edges of the image. Since mean absolute error (MAE) can keep edge information better due to its processing method with large errors [33]. For given N input image sets $\{\mathbf{O}^i, \mathbf{J}^i\}_{i=1}^N$, function 9 need to be minimized as its goal, where $\mathbf{\Theta}$ means all training parameters and $f()$ represents this entire network.

$$\mathcal{L}(\mathbf{\Theta}) = \frac{1}{N}\sum_{i=1}^{N}\|f(\mathbf{J}^i;\mathbf{\Theta}) - \mathbf{O}^i\|_1 \tag{9}$$

To avoid amplify noise, some regular items need to be added of this optimization problem model to maintain the smoothness of the image. Total Variation loss (TV loss) is a commonly used regular item, inspired by the methods of [13], TV Loss is attached into the total loss function to constrain noise as well. For 2D images, The total-variation loss (10) proposed by Rudin at el. [30] is

$$\mathcal{R}_{V^\beta}(\mathbf{x}) = \sum_{i,j}\left((x_{i+1,j} - x_{i,j})^2 + (x_{i,j+1} - x_{i,j})^2\right)^{\frac{\beta}{2}} \tag{10}$$

where $\mathbf{x}$ is the input image, i and j shows the coordinates of one pixel, and in this case β is always equal to 2. By introducing TV loss, spatial smoothness in the generated image can be enhanced. Additionally, when β is larger than one, it will lose some clarity as sacrifice, and since TV loss is a regularization part, the influence coefficient of TV loss on the total loss function is adjusted to 0.1.

3.3 Method details

Parameters setting All convolutional kernels are in 3 by 3 size, and iteration number is 75. Noting that in dynamic convolution, there are three different dilation values: 1,2 and 4, in order to concatenate them in the following step without changing the size of feature maps, the padding values are set to 1, 2 and 4 correspondingly, and the kernel number K set as 3. For training batch, the size is 8, and the testing batch size is 4. The learning rate is fixed to 10^{-4}.

Training and testing data-set There are 200 different JPEG images in both training and testing data set. All the images are from BSD 500 [2], and the training and testing images are non-associated with the resolution of 481×321. To have the multi-quality images, every image is compressed to the 10, 20 and 30 as quality values. And it is worth to emphasize that the training process is conducted toward these three different compressed images in the same model. So in total, there are 600 pairs of image set in each epoch, that are from 200 raw images and its three compressed version counterpart.

Evaluation setting After the testing stage, peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) are used to evaluate the results as two important indicators, which are commonly used objective measurement methods for evaluating image quality.

Given a grayscale image I with a size of $m \times n$ and a noisy image K, the mean square error (MSE) is defined as (11):

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \Big[I(i,j) - K(i,j) \Big]^2 \tag{11}$$

Based on MSE, PSNR (in decibel) defined as (12),

$$PSNR = 10 \cdot \log_{10}(\frac{MAX_I^2}{MSE}) \tag{12}$$

where the MAX_I^2 represents the maximum possible pixel value. For color image with three RGB channels, like in this method, the only different with the monochrome image is while calculating the MSE, should sum all squared value differences.

One drawback of PSNR is that it cannot be precisely the same as the visual quality seen by the human eye because the sensitivity of human vision to errors is not absolute. Its perception results will be affected by many factors [20]. For example, human eye is more sensitive to contrast differences with lower spatial frequencies, and the human eye is more sensitive to brightness contrast differences than chroma. But although PSNR has some uncertainty between human vision and its results, it is still worth using as a picture quality quantification index.

Another measurement is SSIM, which denote the similarity of two images. The definition of SSIM is (13),

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{13}$$

in which μ_x and μ_y are the average of x and y respectively. σ_x^2 and σ_y^2 are the corresponding variance value, and σ_{xy} means the covariance of x and y. $c_1 = (k_1 L)^2$,$c_2 = (k_2 L)^2$ are the two are constants used to maintain stability with L as its pixel values range. By default, $k_1 = 0.01$ and $k_2 = 0.03$. The range of SSIM is from -1 to 1, when the two images are exactly the same, the value of SSIM is equal to 1.

As the realization of the structural similarity theory, the SSIM defines structural information from the perspective of image composition as being independent of brightness and contrast, reflecting the properties of the object structure in the scene, and modelling distortion as a combination of three aspects: brightness, contrast and structure.

Both PSNR and SSIM can reflect the quality of the reconstructed pictures to a certain extent and provide a reference for the generated model evaluation.

4 Experiments

4.1 Ablation comparisons

Compared with Fu. Xueyang et al. work [11], in this proposed network architecture, two different features have been modified.

The first one is replacing dilation convolution with dynamic convolution in the feature extraction part. For the dilation convolution, it uses the same filter in different scales, which can enlarge the contextual area without adding extra parameters [37]. In [11], they adopt this method to address various receptive field problem, that share the same propose with this approach. Also, it concatenate these three dilation layers at the end of getting feature maps from different image qualities.

Another one is adding TV loss in the total loss function in order to restrain the noise further. And as mentioned in [11], compared to MAE, MSE is not struggling to preserve image structures and more tolerant with minor errors, so in this discussed networks, MSE loss is also used as the based loss function.

To distinguish their independent influence on the original network by Fu. Xueyang et al., four experiments were performed: the initial network proposed by Fu. Xueyang et al. in [11], the modified network with only adding TV loss function, the adjusted network with only replacing dilation convolution with dynamic convolution, and the combined network with both of attaching TV loss and switching dynamic convolution. To facilitate the following description, unless otherwise specified, the network **A**, **B**, **C** and **D** represent those four networks described above. Noting that although the network **A** is based on the work of Fu. Xueyang et al., it using different testing image set and training epoch number, so all the comparisons are all among these mentioned networks in the current setting, the result of network **A** could be different with [11].

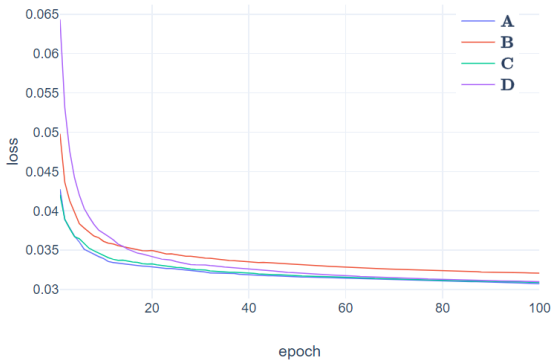


Fig. 2: Traing Loss of four networks

Training Firstly, for the training part, all the networks are trained for the one hundred epochs. Figure 2 indicates the different networks training loss. In this chart, all networks tend to convergence after the 40 epoch, but network **D** has a slower convergence speed compared with the other three. And for the network **B** and network **D**, both of them have added the TV loss, show the higher loss at the first several epoch, because at the original few epochs, the TV loss would give additional errors for the total loss.

Figure 3 and 4 indicate the PSNR and SSIM respectively. From these two charts, there is not obviously difference between these four network only from the training progress. They all have the comparable results with each other.

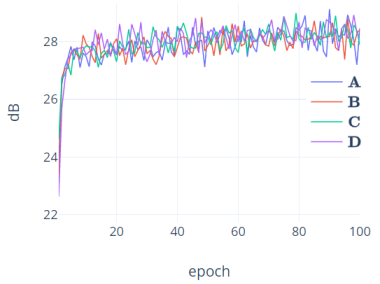


Fig. 3: Training PSNR of four networks

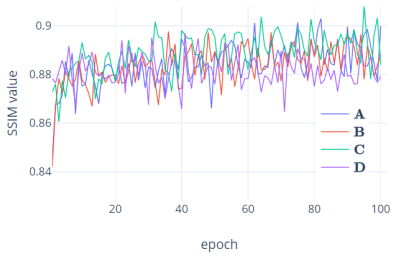


Fig. 4: Training SSIM of four networks

Testing result In testing part, there are 150 images are tested by each network, and for three different image qualities, 50 images are tested under each quality. Also, the PSNR and SSIM are calculated for the quantitative evaluation of each network as well as the compressed JPEG image, which are shown in Table 1. The best result in the certain quality are all in bold (including the overall results), while the lowest values have underlines.

By comparing the testing result among these four networks, each of them gives a better result than the original compressed JPEG image. For PSNR, the network **C** has the best results in all three image qualities, as well as the overall result. Network **D** has a comparable testing PSNR, and with the quality increasing, the gap between those two networks is narrow down. In SSIM assessment, the network **D** has a higher structural similarity with the ground truth than the others except when the image quality is 10, in which the network **C** has a little better than the others.

Among these four networks, network **B** has comparable values in both PSNR and SSIM with network **A**. In the comparisons of different image qualities, when the quality is 10, network **B** does not have better PSNR than network **A**, when the compressed image quality increase, especially when quality is 30, the network

Table 1: Comparison of four networks testing result

Network	Compressed JPEG	**A**	**B**	**C**	**D**
PSNR	quality = 10	27.20195220	27.15529762	**27.49716605**	27.35324233
	quality = 20	29.51739216	29.49505357	**29.92748446**	29.81997779
	quality = 30	30.59214258	30.60956523	**31.06809419**	31.00578163
Overall	27.511688	28.710313	28.704696	**28.936940**	28.918165
SSIM	quality = 10	0.866696935	0.866316018	**0.871134491**	0.869668812
	quality = 20	0.912456598	0.912632115	0.916135771	**0.916801777**
	quality = 30	0.928541638	0.928925026	0.931885152	**0.933535218**
Overall	0.881532	0.902565	0.902624	0.906385	**0.906669**

B overtakes **A** in both PSNR and SSIM. Network **C** and **D** have better performance than those previous two models, while the former has the best overall PSNR result and the latter have the most competitive result in SSIM. The same situation also appeared in these two networks. With the higher image quality, the network **D** is prone to have an accelerated better result than the network **C**, especially in the SSIM. Figure 5 and 6 present the testing result in terms of different image qualities of these four networks.

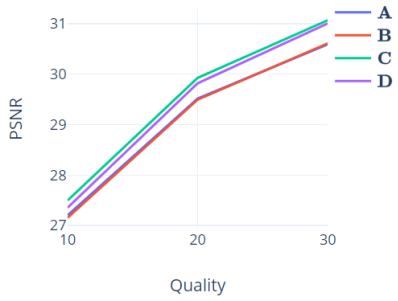


Fig. 5: Testing PSNR of four networks

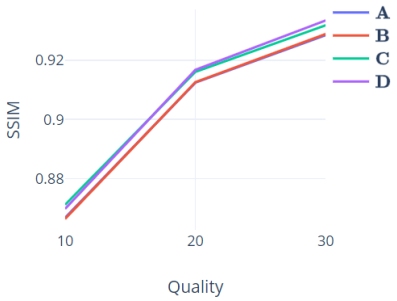


Fig. 6: Testing SSIM of four networks

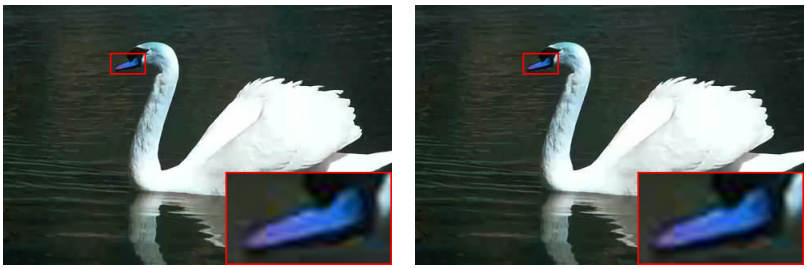
4.2 Reconstructed JPEG image

Figure 7,8 and 9 show the restored image results from BSD500 during the testing session of four networks, and their corresponding PSNR and SSIM are shown below them. The values in bold indicate the highest value among its counterparts. In the low compressed level, the network **C** usually gives better de-blocking result, while the reconstructed images that from high quality dataset, have the better visualization using network **D**.

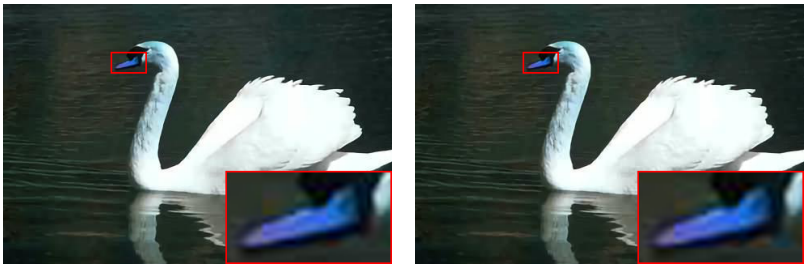
(a) **A**: 26.145846—0.851684 (b) **B**: 26.179224—0.851521

(c) **C**: **28.583516**—**0.880268** (d) **D**: 28.432769—0.879082

Fig. 7: Visual comparison of reconstructed images (quality = 10). The image of network **C** has the straighter and smoother edges.



(a) **A**: 29.434875—0.885535 (b) **B**: 29.551224—0.886465



(c) **C**: **31.263746**—0.927009 (d) **D**: 31.222759—**0.928812**

Fig. 8: Visual comparison of reconstructed images (quality = 20). The colors in images of network **C** and network **D** are more uniform.

(a) **A**: 29.100215—0.910681 (b) **B**: 29.086936—0.911006

(c) **C**: 33.481524—0.945058 (d) **D**: **33.565527—0.947649**

Fig. 9: Visual comparison of reconstructed images (quality = 30). There are less artifacts around the grass in the image of network **D**.

5 Analysis

From the previous experiments, the network **C** and **D** slightly overperform the other two networks, which are using dilation convolution in the feature extraction part. The utilization of dynamic convolution expresses its advantages comprehensively that it has a good capability in solving the multi-task problem in shallow neural networks. Compared with the TV loss, the dynamic convolution contributes more to better performance, while the former modification in this network does not have a distinct impact. However, in the intra-comparison, the TV loss has a greater influence on the high-level compressed images and has the tendency to give a better result when the quality keeps increasing. That is one of the properties of TV loss, which has a trade-off between the artifact reduction and image smoothness.

6 Conclusion

This proposed network, which adopts dynamic convolution, benefited by the combination of sparse coding and the CNN, achieves competitive anti-artifact performance. The multi-scale feature extraction method with DCSC approach is straightforward and well-structured. The introducing of concatenated dynamic convolution allows the single lightweight model to handle the different compression qualities at the same time. Typically, in the high compressed level, this

network gives a more noticeable result due to the added TV loss part. Overall, this network has a good explainability and adaptability on image artifacts reduction.

For the future work, the consolidation of DCSC and deep learning with dynamic convolution has the potential ability to deal with the other restoration tasks. For instance, the DCT domains can be further explored using this model.

References

1. Ahmed, N., Natarajan, T., Rao, K.: Discrete Cosine Transform. IEEE Transactions on Computers. C-23, 90-93 (1974).
2. Arbeláez, P., Maire, M., Fowlkes, C., Malik, J.: Contour Detection and Hierarchical Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 33, 898-916 (2011).
3. Chen, Y., Dai, X., Liu, M., Chen, D., Yuan, L., & Liu, Z.: Dynamic convolution: Attention over convolution kernels. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition on Proceedings, pp. 11030-11039 (2020).
4. Chen, Y., Pock, T.: Trainable Nonlinear Reaction Diffusion: A Flexible Framework for Fast and Effective Image Restoration. IEEE Transactions on Pattern Analysis and Machine Intelligence. 39, 1256-1272 (2017).
5. Chung-Bin, W., Bin-Da, L., Jar-Ferr, Y.: Adaptive postprocessors with DCT-based block classifications. IEEE Transactions on Circuits and Systems for Video Technology. 13, 365-375 (2003).
6. Connell, J.: A Huffman-Shannon-Fano code. Proceedings of the IEEE. 61, 1046-1047 (1973).
7. Dong, C., Deng, Y., Loy, C. C., & Tang, X.: Compression artifacts reduction by a deep convolutional network. In : IEEE International Conference on Computer Vision on Proceedings, pp.576-584 (2015).
8. Dong, C., Loy, C., He, K., Tang, X.: Learning a Deep Convolutional Network for Image Super-Resolution. Computer Vision – ECCV 2014. 184-199 (2014).
9. Dyer, E., Johnson, D., Baraniuk, R.: Learning modular representations from global sparse coding networks. BMC Neuroscience. 11, (2010).
10. Foi, A., Katkovnik, V., Egiazarian, K.: Pointwise Shape-Adaptive DCT for High-Quality Denoising and Deblocking of Grayscale and Color Images. IEEE Transactions on Image Processing. 16, 1395-1411 (2007).
11. Fu, X., Zha, Z. J., Wu, F., Ding, X., & Paisley, J.: Jpeg artifacts reduction via deep convolutional sparse coding. In: IEEE/CVF International Conference on Computer Vision on Proceedings, pp. 2501-2510 (2019).
12. Galteri, L., Seidenari, L., Bertini, M., Bimbo, A.: Deep Universal Generative Adversarial Compression Artifact Removal. IEEE Transactions on Multimedia. 21, 2131-2145 (2019).
13. Gatys, L. A., Ecker, A. S., & Bethge, M.: Image style transfer using convolutional neural networks. In: IEEE conference on computer vision and pattern recognition on Proceedings, pp. 2414-2423 (2016).
14. Guangtao, Z., Wenjun, Z., Xiaokang, Y., Weisi, L., Yi, X.: Efficient Deblocking With Coefficient Regularization, Shape-Adaptive Filtering, and Quantization Constraint. IEEE Transactions on Multimedia. 10, 735-745 (2008).
15. Hannuksela, M., Lainema, J., Malamal Vadakital, V.: The High Efficiency Image File Format Standard [Standards in a Nutshell]. IEEE Signal Processing Magazine. 32, 150-156 (2015).

16. Heide, F., Heidrich, W., & Wetzstein, G.: Fast and flexible convolutional sparse coding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition on Proceedings, pp. 5135-5143, (2015).
17. Hinds, S., Fisher, J., D'Amato, D.: A document skew detection method using run-length encoding and the Hough transform. [1990] Proceedings. 10th International Conference on Pattern Recognition.
18. Howard, P., Vitter, J.: Fast and efficient lossless image compression. [Proceedings] DCC '93: Data Compression Conference.
19. Hu, J., Shen, L., Albanie, S., Sun, G., Wu, E.: Squeeze-and-Excitation Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence. 42, 2011-2023 (2020).
20. Huynh-Thu, Q., Ghanbari, M.: Scope of validity of PSNR in image/video quality assessment. Electronics Letters. 44, 800 (2008).
21. JPEG - JPEG, https://jpeg.org/jpeg/. Last accessed 26 Nov 2020
22. Khayam, S. A.:"The discrete cosine transform (DCT): theory and application. Michigan State University 114, 1-31 (2003).
23. Knuth, D.: Dynamic huffman coding. Journal of Algorithms. 6, 163-180 (1985).
24. Krizhevsky, A., Sutskever, I., Hinton, G.: ImageNet classification with deep convolutional neural networks. Communications of the ACM. 60, 84-90 (2017).
25. Li, Y., Guo, F., Tan, R. T., & Brown, M. S.: A contrast enhancement framework with JPEG artifacts suppression. European conference on computer vision, pp. 174-188. Springer, Cham (2014).
26. List, P., Joch, A., Lainema, J., Bjontegaard, G., Karczewicz, M.: Adaptive deblocking filter. IEEE Transactions on Circuits and Systems for Video Technology. 13, 614-619 (2003).
27. Liu, X., Wu, X., Zhou, J., Zhao, D.: Data-Driven Soft Decoding of Compressed Images in Dual Transform-Pixel Domain. IEEE Transactions on Image Processing. 25, 1649-1659 (2016).
28. Nagy, J., O'Leary, D.: Restoring Images Degraded by Spatially Variant Blur. SIAM Journal on Scientific Computing. 19, 1063-1082 (1998).
29. Redmill, D., Kingsbury, N.: The EREC: an error-resilient technique for coding variable-length blocks of data. IEEE Transactions on Image Processing. 5, 565-574 (1996).
30. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena. 60, 259-268 (1992).
31. Rothe, R., Timofte, R., Van Gool, L.: Efficient regression priors for reducing image compression artifacts. 2015 IEEE International Conference on Image Processing (ICIP). (2015).
32. Tai, Y., Yang, J., Liu, X., & Xu, C.: Memnet: A persistent memory network for image restoration. In: IEEE international conference on computer vision on Proceedings, pp. 4539-4547 (2017).
33. Tuchler, M., Singer, A., Koetter, R.: Minimum mean squared error equalization using a priori information. IEEE Transactions on Signal Processing. 50, 673-683 (2002).
34. Usevitch, B.: A tutorial on modern lossy wavelet image compression: foundations of JPEG 2000. IEEE Signal Processing Magazine. 18, 22-35 (2001).
35. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image Quality Assessment: From Error Visibility to Structural Similarity. IEEE Transactions on Image Processing. 13, 600-612 (2004).

36. Wang, Z., Liu, D., Chang, S., Ling, Q., Yang, Y., & Huang, T. S.: D3: Deep dual-domain based fast restoration of JPEG-compressed images. In: IEEE Conference on Computer Vision and Pattern Recognition on Proceedings, pp. 2764-2772 (2016).
37. Yu, F. and Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv preprint, arXiv:1511.07122 (2015).
38. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. IEEE Transactions on Image Processing. 26, 3142-3155 (2017).
39. Zhang, X., Yang, W., Hu, Y., & Liu, J.: DMCNN: Dual-domain multi-scale convolutional neural network for compression artifacts removal. 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 390-394. IEEE (2018).