Combined Convolutional Neural Network for Highly Compressed Images Denoising

Binying LIU
Graduate School of Information,
Production and Systems,
Waseda University
Email:binying-lliu@moegi.waseda.jp

Sei-ichiro KAMATA
Graduate School of Information,
Production and Systems,
Waseda University
Email:kam@waseda.jp

Abstract—Many methods for denoising additive white Gaussian images have been developed, such as the use of non-local mean filters (NLF) and deep convolutional neural networks (CNN). However, these denoising methods still have many limitations on compressed images such as JPEG2000 compression. Based on quantization of noisy wavelet coefficients, JPEG2000 may lead to very specific visual artifacts. This compressed image's noise distribution model is highly spatially correlated and very different from the noise distribution model in additive Gaussian white noise images. In this paper, we propose a convolutional neural network structure combined with nonlocal filter. At first a convolutional neural network have been trained by using highly compressed noisy images to obtain a specific noise model estimation and this noise model estimation is used for the residual neural network. Secondly, it based on non-proximity average filtering, where a similar block selection method is modified to find block artifacts in the compressed image and then do denoising. Finally, combining these two methods can get a clear image output. The evaluation results of this method on the grayscale image dataset are better than the latest technology.

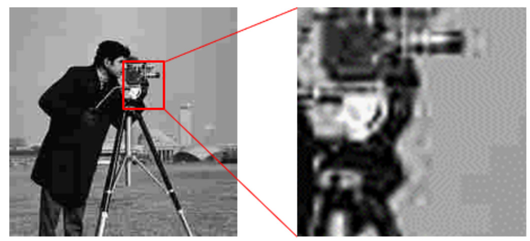
Contribution— We produced a noise distribution CNN model that can predict the noise of highly compressed images with complex noise distribution, and combine CNN and Non-local mean filters to obtain good denoising results.

Index Terms—image denoising; highly compressed image; convolu-tional neural network; nonlocal filters; BM3D.

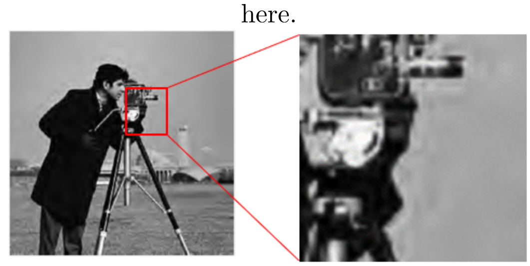
I. Introduction

In digital images, image denoising is the process of reducing noise, which is a very important issue in image processing and computer vision. We use the denoising method in image, which can increase the quality of the output image, and the clean image obtained can be better used in other image vision fields, such as object segmentation, demosaicing, classification, and recognition [1].For decades, many images denoising researches in this area are all based on the research on noise pictures with additive white Gaussian noise (AWGN) [2]–[6], and achieved good performance results [7]–[9].

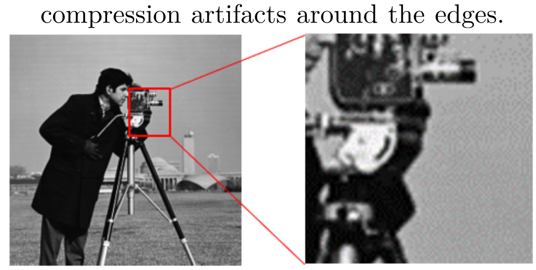
Lossy compression for image, such as JPEG2000, WebP, and HEVC are encoding methods for digital images. Compared with additive Gaussian white noise, the noise in compressed image is generated from the undesired complex compression artifacts while using the lossy compression method. The image coding compression method uses linear reversible transformation (such as JPEG compressed block-by-block DCT or JPEG2000 wavelet transform), and then quantizes the transformed image data matrix, and finally encodes it by a lossless encoder.



(a) The JPEG-compressed image with the Quality parameter equal to 10. At the edges of the camera and tripod, we can find many compression artifacts existing here.



(b) The JPEG2000 compressed image with a compression ratio equal to 20. There are many compression artifacts around the edges.



(c) The restored image by the proposed method. Here we can find that we remove some compression artifacts and preserve information at the edges of the image.

Fig. 1. Example of higly-compressed images using JPEG.

Take JPEG compression method as an example. JPEG

978-1-7281-9331-1/20/$31.00 ©2020 IEEE

is a classic highly compressed encoding and decoding method. The first step in JPEG compression is to divide the image into 88 pixel-level blocks, which are processed separately during the entire compression process. If this image is an RGB model, usually we need to convert it to a YCbCr model for processing. An important part in the JPEG compression process is applying discrete cosine transformation (DCT) on each block, which can decompose an array into the sum of several arrays, while JPEG2000 is using a wavelet transform. In the JPEG compression process, we perform a two-dimensional DCT transformation on each 88 pixel-level block to obtain an 8×8 floating-point number matrices. In order to use less space to store these floating-point numbers, the quantization matrix calculation is added to obtain less valid data. The quantization process results in a loss of image information, as shown in Fig. 1(a). This step will cause many different artifacts shown in the conpressed images. When encoding each block, if the correlation with neighboring blocks is not considered, block artifacts will appear, resulting in discontinuities in the 88 borders [10].

In recent years, people have begun to use CNN for image noise reduction. Starting from noise pictures with additive white Gaussian noise (AWGN), many researchers have developed an excellent method using CNN [11], [12]. As indicated by [13], depending on when the distribution of synthetic noise and real noise are well matched, the CNN denoiser can generally get good results. According to [14], [15], however, AWGN is a simple and tractable mathematical models which are useful for adding noise into the noise-free image before these other situations are considered Therefore, using CNN denoisers, just as DnCNN and FFDNet, it is easy to generate residual models that conform to the AWGN distribution.

Non-local filter (NLF) is also widely used in the field of denoising the image which added with AWGN. Currently, the most effective trainditional methods for image denoising can be roughly categorized into nonlocal filter, such as NLBayes [16] and Block Matching 3D (BM3D) [17]. NLF can combine similar image blocks for collaborative filtering, so that when the image shows strong self-similarity (for example, on edges or regular textures), excellent noise removal results can be obtained. On the other hand, when the image has weak self-similarity, the denoising performance may not be well.

In this paper, the framework has been proposed that uses a combination of NLF and CNN filter to reconstruct highly compressed images, such as JPEG-compressed images. For CNN filter, our method is depending on the residual learning, where the input is noise image, and CNN is used to estimate the noise distribution of the noise images. Then we created JPEG-compressed noise model and use more than 400 different kinds of gray images train the CNN filter. To enlarge the training dataset, we use the data augmentation method. Using down-sampling and up-sampling to adjust the size and number of input images, which can increase the training speed. But CNN filter presents bad performance on regular texture with great self-similarity, while NLF can overcome this problem. At the same time, we added the nonlocal filter to enhance the detail information for the output denoised image. After doing the experiences, our method can do a great job in denoising for highly JPEG-compressed images.

II. Related Work

A. Deep CNN Filters

The emergence of CNN provides a great space for development in image processing and calculation. On this basis, deep neural networks (DNNs) have also been widely used in image denoising technology. In the early days, most deep neural networks could not get the state-of-the-art denoising results [18] [19] [20], until Burger et al. [21]. Subsequently, CSF [22] and TNRD [5] developed optimization algorithms for solving the field of expert models to learn the segmented inference program. Aiming at the noise image denoising generated by the noise superposition on the clean image, Zhang et al. [11] [11] proposes a denoising CNN, which combines residual learning [23] and batch normalization [24], and its denoising performance is better than the traditional denoising method. Noise2Noise [25] also achieves state-of-the-art, while the clean images are not used.

CNN has powerful modeling capabilities, and the studies [11] [26] [27] show that for blind Gaussian denoising it is possible to get a single model. It's easy to find that this kind of CNN denoised models may show the over-fitting results to AWGN and can not deal with more complex noise models such as JPEG-compressed noise. While the noised distribution is known, non-blind CNN denoisers can achieve great results on JPEG-compressed noise image in this case. To this research, the CNN filter from my structure contains a noise estimation subnetwork to do the noise estimation using an asymmetric loss for JPEG images.

B. Nonlocal means Filter for Denoisers

Non-local means is a traditional denoising algorithm used in image processing. Different from the local means filters, non-local means filters take a mean value of all pixels in one complete image, then weighted by calculating the similarity between these pixels in one image block to the target block. At the same time, local means filters calculate the mean of a group of pixels surrounding the target pixel to filter the noise of the image. Compared with local mean algorithms, these results show the greater filtering results and less loss of detail features in this image.

There existing many methods based on nonlocal filter have been used to eliminate AWGN, such as NLMeans, NLBayes [16] and Block Matching 3D (BM3D) [17]. The first step in NLFilter is block matching (BM). BM stands for finding a group of similar blocks. The second step is to combine similar blocks and obtain the denoised pixel blocks by averaging the surrounding pixels. NLMeans uses the entire image to denoise. Taking the image block as the unit, after the similar areas in the image are found, the similar blocks are averaged, which can better eliminate the Gaussian noise in the image. By evaluating for each group of similar patches a Gaussian vector model, the NL-Bayes strategy improves on NL-Means. In this case, the type and distribution of noise negatively impact the NLF.

In this case, the type and distribution of noise negatively impacts the NLF. It is effective for JPEG-compressed noise reduction. Because the noise distribution of compressed images is more complicated than the AWGN images. Compare with NL-Means and NL-Bayes, BM3D can achieve better results in terms of denoising compressed images, by flexibly changing the hard threshold.

C. Denoising for highly compressed images

For highly compressed image denoising, most of existing algorithms are deblocking oriented or restoration oriented methods. Because the JPEG compression method is block compression, removing ringing artifacts and blocks become the target for deblocking oriented. There are different kinds of denoising filters [28] [29] [30] in the spatial domain, and these denoising filters have been used to deal with blocking artifacts in specific areas (such as smooth regions, texture, and edge). And in the frequency domain, Liew et al. [31] used derive thresholds and wavelet transform for highly compressed image denoising.

The Pointwise Shape-Adaptive DCT (SA-DCT) [32] is usually used for the deblocking oriented method as an excellent method. SA-DCT firstly performs a one-dimensional DCT calculation for each column of pixels, and then collects the DCT coefficients with the same subscript to perform one-dimensional DCT calculation. The resulting DCT coefficients are located in the MM size block, and the number is the same as the number of pixels in the original block. The DC coefficients have been concentrated in the upper left part of the block. When decoding, the original image data is restored by combining the shape information.

However, it's difficult for SA-DCT transform to make a good division of natural image information and noise. It pushes the pixels to be flush with one side of the rectangular border, so some spatial correlations may be lost. In this way, column DCT transformation will have greater distortion. And for this method, it may cause important high-frequency parts to merge with the boundary part. And the distribution coefficients may not have the same phase with each other, which will produce the discontinuities in the second direction wavelet decomposition.

III. Proposed Method

Our proposed framework is presented in this section, shown like Fig. 2. The whole structure is divided into two parts: CNN filter and NLF.

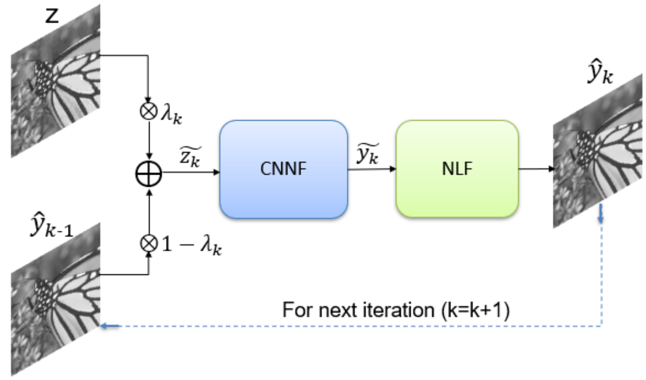


Fig. 2. The k-th iteration of the proposed framwork.

A. CNN Filter

CNN is widely regarded as a tool that can learn features from large amounts of training data. In recent years, CNN has been successfully used in noise extraction to reduce the noise of images with additive Gaussian white noise, such as DnCNN and FFDNet. But these two methods cannot be used in a JPEG-compressed image because of compression artifacts which distribution is quite different from the Gaussian distribution. Therefore, training a CNN denoiser using JPEG-compressed images should train a CNN filter with excellent performance. Usually, the additive white Gaussian noise (AWGN) can be added into a clean images:

$$z = y + \eta, \quad \eta(\cdot) \sim N(0, \sigma^2), \tag{1}$$

while y is a clean image, z is a noisy image and σ is the known noise standard deviation. Different from AWGN, the distribution of noise contributed from JPEG compression is not so easy to get to know. The JPEG compression artifact observation model is:

$$z = JPEG(y, QP), \tag{2}$$

in this case, quantization parameters (QP) are used to generate the JPEG noisy images for training at different quality level.

As shown in Fig. 3, the CNN structure contents two different parts. The first part is used for noise estimation, it can also call CNN_I. Different kinds of JPEG-compressed images are the input, and the output is the estimated noise level map $\hat{\sigma}(z) = f_I(z; W_I)$, while W_I is the network parameters of CNN_I. The second part is used for image denoising. This part is a non-blind denoising subnetwork CNN_I, which the input includes both the output $\hat{\sigma}(z)$ of CNN_I and noisy images z. Then the final denoising result is $\hat{y} = f_I(z; \hat{\sigma}(z); W_I)$, while W_I is the network parameters of CNN_I.

The structure of CNN_I is five-layer fully convolutional network plus one-layer down-sampling without pooling and patch normalization operations. The initial input is a full-size noise image. After five-layer fully convolutional network, the output is the same size noise estimation feature map $\hat{\sigma}(z)$. Then the number of channels is 64

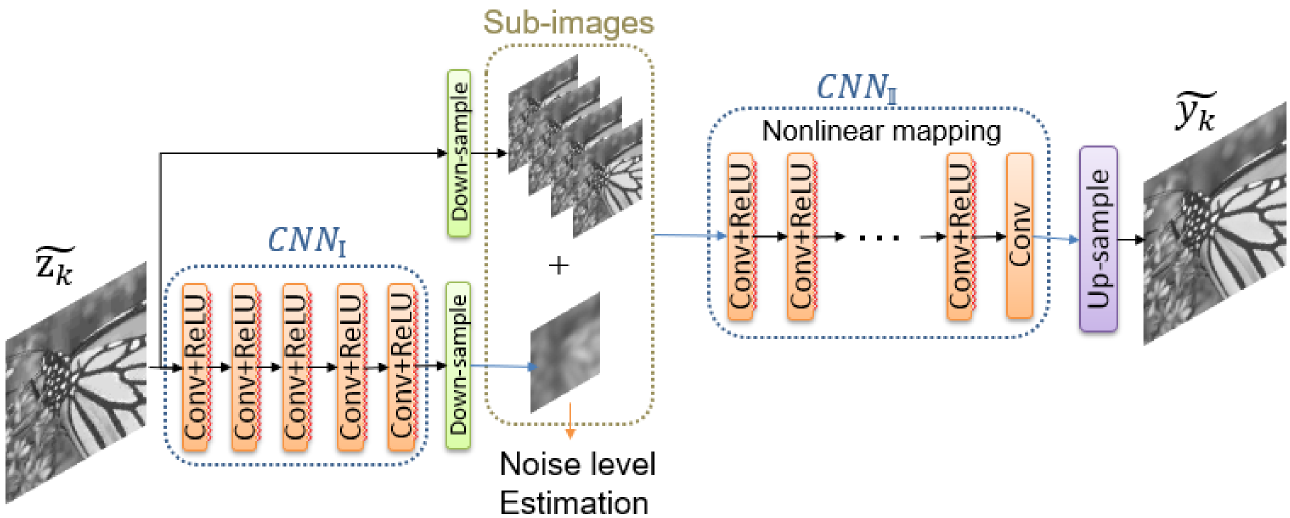


Fig. 3. Illustration of our CNN filter for denoising of JPEG-compresed image.

and the size of the filter is 33 in each convolution layer. There will be a ReLU layer after each convolutional layer [37]. As for CNN_{II}, we use a nonlinear mapping method. Both noise image z and noise estimation feature map $\hat{\sigma}(z)$ become the input of CNN_{II}, then a prediction of the denoised image $\hat{y}$is given as an output. In the CNN_{II}, the key ideal is using residual learning to get the residual mapping result $R_{II}(z; \hat{\sigma}(z); W_{II})$. The prediction of the denoised image $\hat{y} = z + R_{II}(z; \hat{\sigma}(z); W_{II})$. The structure of CNN_{II} is given in Fig. 3, where the down-sampling layer is in the first layer and the last layer is an up-sampling layer. This time, we adopted fifteen-layer fully convolutional network without pooling and patch normalization operations, which all the size of the filter is 33 and number of channels is 64. In this case, we can find that using downsampling can reduce the size of pictures passing through the network, which can effectively reduce the time for running the code.

In order to obtain a satisfying CNN filter, we use the loss function to measure the training result. CNN_I is a noise estimation subnetwork. After being trained, the output will be a noise prediction map for highly compressed pictures. Then the noise prediction map and the noise image are the input to the non-blind denoising subnetwork CNN_{II}. For non-blind denoising subnetwork CNN_{II}, if in the input, the noise prediction value is bigger than the real noise value (that is, the high estimation error is too large), a better noise-reduced image will be the output.

For example, there is a real noise image noise reduction method FFDnet, whose noise reduction effect is related to the degree of matching between input noise and real noise during training. When the input noise is lesser than the real noise in one noisy image, the noise reduction result will contain more noise. If the noise is bigger than the authenticity in the input, FFDnet outputs a satisfactory noise-reduced image with some details lost. It can achieve the best effect only when the input noise matches the actual noise. In order to take advantage of the asymmetric sensitivity in blind noise reduction, the asymmetric loss can be proposed in the noise estimation subnetwork to reduce underestimation errors on the noise level graph. Used in CNN_I, this asymmetric loss can be used to control the quality of the estimated noise feature map. The estimated noise $\hat{\sigma}(z_i)$ and the ground-truth $\sigma(z_i)$ at pixel i is given. If $\hat{\sigma}(z_i) < \sigma(z_i)$, there must be more training losses. Thus, the asymmetric loss CNN_I is defined as,

$$L_{asymm} = \sum_i \mid \alpha - II_{(\hat{\sigma}(y_i)-\sigma(z_i))<0} \mid \cdot (\hat{\sigma}(y_i) - \sigma(z_i))^2, \tag{3}$$

where $II_e = 1$ for e<0 and 0 otherwise. We set an underestimation error σ, with parameter values ranging from 0 to 0.5. Using under-estimation error, the denoising model is well extended to JPEG compression noise, after imposing a penalty. For the output of denoised image in CNN_{II}, the reconstruction loss is defined as,

$$L_{res} = \parallel \hat{y} - y \parallel_2^2 . \tag{4}$$

Then, the overall loss function of the CNN filter is,

$$L = L_{res} + \lambda_{asymm} L_{asymm}, \tag{5}$$

where λ_{asymm} shows the parameters of tradeoff for the asym-metric loss.

In our work, the noise model is used given by MATLAB software such as Eqn.(2), to generate JPEG compressed noise images. The training dataset includes 400 clean gray images, which size is 180180 in pixel level. This dataset includes many pictures of different animals, plants and the environment, according to [35], [38], [39][17,18,19]. Specifically, in order to increase the number of training

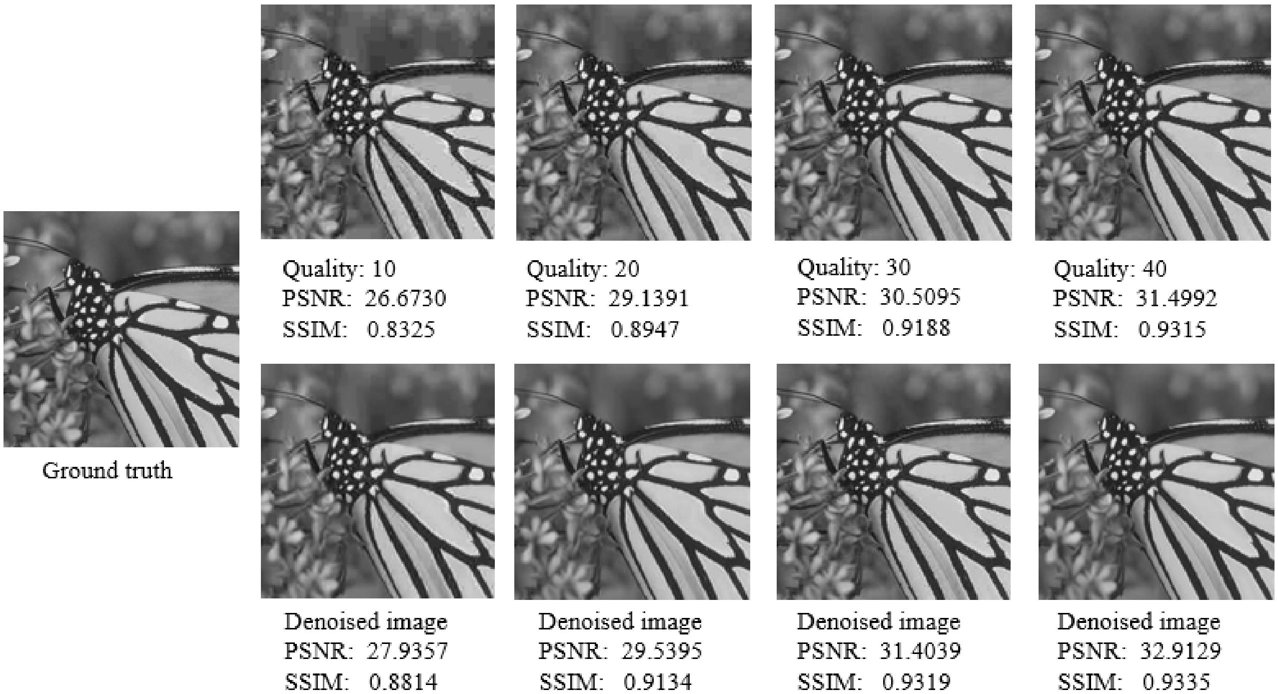


Fig. 4. Denoising reaults for JPEG-compressed images with different compression qualities.

images, we use the data augmentation method, which is to rotate or flip the clean image randomly. Then different quantization parameters (QP) are randomly generated and used to calculate the JPEG-compressed noise images for training. We use this batch of synthetic images to train the CNN filter and update the coefficients in the convolutional layer to minimize all the losses in Eqn.(5). We have found from experience that this neural network training method can obtain effective denoising quality for JPEG compressed images.

B. Non-Local Filter

We used the BM3D as nonlocal filter. The process of BM3d includes two steps.

The first step is basic estimation. We selected some reference blocks of equal size in the noisy image. Usually 3 pixels are selected for a step size, and the complexity is reduced to 1/9). Search within the surrounding area of the reference block to find N blocks with the least degree of difference, and integrate these blocks into a 3-dimensional matrix $G(S) = \{S_1, \cdots, S_N\}$, where S_i is grouping for one similar block. Each S_i contains the coordinates of N_2 blocks of size $N_1 \times N_1$ similar to each other. The order of integration has little effect on the result. Then, the reference block will be integrated into the 3-dimensional matrix, and difference of the degree is equal to 0. The process of finding similar blocks can be shown as

$$G(S) = \{Q : d(S_i, S_j) \leq \Gamma_{step1}\}. \tag{6}$$

While S_j is reference block, S_i is similar block and $d(S_i, S_j)$ represents the Euclidean distance between two blocks.

Then we do collaborative filtering to the 3-dimensional matrix $S = \{S_1, \cdots, S_N\}$. For each S_i, we use wavelet transform to do the 2-D transform, and for the third dimension, we use the Haar wavelet transform. The filtered group is:

$$Q_{estimated}(S) = \Gamma_{2D}^{-1}(\Gamma_{1D}^{-1}(\gamma(\Gamma_{1D}(\Gamma_{2D}(Q(S)))))), \tag{7}$$

while γ is a hard threshold operation. If the input of γ is less than noise parameter, all the input values become 0; otherwise, the output is the same as the input. Then do the aggregation to all basic estimated blocks.

The second step is final estimation. After grouping we can get two different 3-dimensional matrixes $G(S)$ and $Q_{estimated}(S)$. During the collaborative filtering, let $\Gamma_{3D}(Q(S)) = \Gamma_{1D}(\Gamma_{2D}(Q(S)))$, the filtered group is:

$$Q_{last}(S) = \Gamma_{3D}^{-1}(W(\Gamma_{3D}(Q(S)))), \tag{8}$$

while W is a parameter from wiener filtering. And after aggregation, we can get the final estimated denoised images.

C. Combination of CNNF and NLF

The whole structure is shown as Fig. 2. Like the testing process, the training process is iterative. In the $k-th$ iteration of the framework, first combine the previous estimates of the JPEG compressed image z and the previous estimate $\hat{y}_{k-1}$ into the input noise image of the CNN filter $\tilde{z}_k = \lambda_k z + (1-\lambda_k)\hat{y}_{k-1}$. Then, by using the convolutional neural network as a filter (CNNF), the output $\tilde{y}_k$ is the result from the input of CNNF $\tilde{z}_k$. In the end, processed by the Non-local filter (NLF), the new estimate $\hat{y}_k$ is constituted by the output of the NLF. This method iterates with $1 = \lambda_1 > \cdots > \lambda_{k-1} > \lambda_k > \cdots > 0$.

IV. Experimental Results

We used the train400 dataset [11], [12] to train my CNN filter. This dataset includes 400 clean images. We use the standard JPEG compression method just like the other compression artifacts reduction methods, and use the JPEG quality randomly setting QP from 5 to 50 (from very bad quality to great quality) in MATLAB JPEG encoding and decoding.

To reduce the training time, we use down-sampling to sample all input images, including the noise estimation feature map. The size of the full image is 180180, and after down-sampling, the size of input become 9090 sub-images. After doing a rotation or flipping, the sub-images could provide 13300 training samples.

The Set12 dataset is used as test set to get the results about denoising performance. This dataset includes 12 gray clean images. This is widely used in the denoising assessment for different kinds of noisy images. To evaluate the comprehensive denoising qualitative for the highly compressed images, we apply the peak signal to noise ratio (PSNR) and structural similarity (SSIM) for denoisied image's quality assessment.

TABLE I: THE MEAN VALUES OF PSNR AND SSIM TESTED ON THE SET12 DATASET

Eval.Mat	Quality	JPEG	SA-DCT	CCNN (ours)
PSNR	10	27.65	28.32	28.79
	20	30.01	30.63	31.02
	30	31.37	31.98	32.35
	40	32.32	32.79	32.99
SSIM	10	0.7960	0.8044	0.8204
	20	0.8617	0.8713	0.8863
	30	0.8898	0.9059	0.9101
	40	0.9062	0.9194	0.9309

We compare our method (CCNN) with the SA-DCT [32] [30] method, which is widely used for compression artifacts reduction as the state-of-the-art method. The PSNR and SSIM results are listed in the Table I testing on the Set12 dataset. We choose four different quantization parameters (QP), which also means Quality in this case. Noted that our method achieves a better result than SA-DCT. The noise reduction capability of our method (CCNN) also has been shown in Fig. 4 for different compression qualities.

These experiments were finished on a computer using Ubuntu 16.04 LTS, which also equipped with an Asus GeForce GTX 1080 GPU.

V. Conclusion

We use a combination of CNN and traditional nonlocal means filters to achieve good results in noise reduction of highly compressed image, such as JPEG-compressed images. The superiority of this method is reflected in two aspects. The first is our breakthrough use of a JPEG-compressed image and clean image combination as a noise-clean image pair to train CNNs. The trained estimated noise model can well match the actual JPEG noise model. This method can also be used to train a JPEG2000 noise distribution model. Second, we have improved the parameters of hard thresholding in BM3D, which is beneficial to the identification of noise artifacts caused by highly lossy compression. We can achieve effective noise reduction for highly compressed images. In the further work, we want to extend this work on denoising for H.265 compression image, which is the latest high efficiency video coding method.

References

[1] P. Milanfar, "A tour of modern image filtering: New insights and methods, both practical and theoretical," IEEE Signal Processing Magazine, vol. 30, no. 1, pp. 106–128, Jan 2013.

[2] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," Signal Processing, IEEE Transactions on, vol. 54, pp. 4311 – 4322, 12 2006.

[3] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Color image denoising via sparse 3d collaborative filtering with grouping constraint in luminance-chrominance space," vol. 1, 09 2007, pp. I – 313.

[4] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2862–2869.

[5] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1256–1272, 2017.

[6] U. Schmidt and S. Roth, "Shrinkage fields for effective image restoration," in 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2774–2781.

[7] P. Chatterjee and P. Milanfar, "Is denoising dead?" IEEE Transactions on Image Processing, vol. 19, no. 4, pp. 895–911, 2010.

[8] A. Levin, B. Nadler, F. Durand, and W. Freeman, "Patch complexity, finite pixel correlations and optimal denoising," vol. 7576, 10 2012, pp. 73–86.

[9] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (red)," SIAM Journal on Imaging Sciences, vol. 10, 11 2016.

[10] C. Dong, Y. Deng, C. C. Loy, and X. Tang, "Compression artifacts reduction by a deep convolutional network," in 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 576–584.

[11] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," IEEE Transactions on Image Processing, vol. 26, no. 7, pp. 3142–3155, 2017.

[12] K. Zhang, W. Zuo, and L. Zhang, "Ffdnet: Toward a fast and flexible solution for cnn-based image denoising," IEEE Transactions on Image Processing, vol. 27, no. 9, pp. 4608–4622, 2018.

[13] C. Martin and M. Mahoney, "Rethinking generalization requires revisiting old ideas: statistical mechanics approaches and complex learning behavior," 10 2017.

[14] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian, "Practical poissonian-gaussian noise modeling and fitting for single-image raw-data," IEEE Transactions on Image Processing, vol. 17, no. 10, pp. 1737–1754, 2008.

[15] T. Plötz and S. Roth, "Benchmarking denoising algorithms with real photographs," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2750–2759.

[16] M. Lebrun and J.-M. Morel, “Implementation of the ”non-local bayes” (nl-bayes) image denoising algorithm,” Image Processing On Line, vol. 3, pp. 1–42, 06 2013.
[17] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-d transform-domain collaborative filtering,” IEEE Transactions on Image Processing, vol. 16, no. 8, pp. 2080–2095, 2007.
[18] V. Jain and H. Seung, “Natural image denoising with convolutional networks.” 01 2008, pp. 769–776.
[19] T. Rabie, “Robust estimation approach for blind denoising,” IEEE transactions on image processing : a publication of the IEEE Signal Processing Society, vol. 14, pp. 1755–65, 12 2005.
[20] J. Xie, L. Xu, and E. Chen, “Image denoising and inpainting with deep neural networks,” Advances in Neural Information Processing Systems, vol. 1, 01 2012.
[21] H. Burger, C. Schuler, and S. Harmeling, “Image denoising: Can plain neural networks compete with bm3d?” 06 2012, pp. 2392–2399.
[22] U. Schmidt and S. Roth, “Shrinkage fields for effective image restoration,” 06 2014, pp. 2774–2781.
[23] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 06 2016, pp. 770–778.
[24] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” 02 2015.
[25] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila, “Noise2noise: Learning image restoration without clean data,” 03 2018.
[26] X.-J. Mao, C. Shen, and Y.-B. Yang, “Image denoising using very deep fully convolutional encoder-decoder networks with symmetric skip connections,” 03 2016.
[27] Y. Tai, J. Yang, X. Liu, and C. Xu, “Memnet: A persistent memory network for image restoration,” in 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 4549–4557.
[28] P. List, A. Joch, J. Lainema, G. Bjøntegaard, and M. Karczewicz, “Adaptive deblocking filter,” Circuits and Systems for Video Technology, IEEE Transactions on, vol. 13, pp. 614 – 619, 08 2003.
[29] H. Reeve and Jae Lim, “Reduction of blocking effect in image coding,” in ICASSP ’83. IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 8, 1983, pp. 1212–1215.
[30] C. Wang, J. Zhou, and S. Liu, “Adaptive non-local means filter for image deblocking,” Signal Processing: Image Communication, vol. 28, p. 522–530, 05 2013.
[31] A. W. . Liew and Hong Yan, “Blocking artifacts suppression in block-coded images using overcomplete wavelet representation,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 14, no. 4, pp. 450–461, 2004.
[32] A. Foi, V. Katkovnik, and K. Egiazarian, “Pointwise shape-adaptive dct for high-quality denoising and deblocking of grayscale and color images,” IEEE Transactions on Image Processing, vol. 16, no. 5, pp. 1395–1411, 2007.
[33] Y. Yang, N. Galatsanos, and A. Katsaggelos, “Projection-based spatially-adaptive reconstruction of block-transform compressed images,” 06 1998.
[34] D. Sun and W. Cham, “Postprocessing of low bit-rate block dct coded images based on a fields of experts prior,” IEEE Transactions on Image Processing, vol. 16, no. 11, pp. 2743–2751, 2007.
[35] C. Jung, L. Jiao, H. Qi, and T. Sun, “Image deblocking via sparse representation,” Signal Processing: Image Communication, vol. 27, p. 663–677, 07 2012.
[36] U. Schmidt, J. Jancsary, S. Nowozin, S. Roth, and C. Rother, “Cascades of regression tree fields for image restoration,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, 04 2014.
[37] G. Hinton, “Rectified linear units improve restricted boltzmann machines vinod nair,” vol. 27, 06 2010, pp. 807–814.
[38] S. Nam, Y. Hwang, Y. Matsushita, and S. J. Kim, “A holistic approach to cross-channel image noise modeling and its application to image denoising,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1683–1691.
[39] A. Abdelhamed, S. Lin, and M. S. Brown, “A high-quality denoising dataset for smartphone cameras,” in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 1692–1700.