DCTResNet: Transform Domain Image Deblocking for Motion Blur Images

Paras Maharjan[1], Ning Xu[2], Xuan Xu[2], Yuyan Song[2], Zhu Li[1]
[1]University of Missouri-Kansas City, USA
[2]Kwai Inc., CA, USA

2021 International Conference on Visual Communications and Image Processing (VCIP) | 978-1-7281-8551-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/VCIP53242.2021.9675411

***Abstract*—Pixel recovery with deep learning has shown to be very effective for a variety of low-level vision tasks like image super-resolution, denoising, and deblurring. Most existing works operate in the spatial domain, and there are few works that exploit the transform domain for image restoration tasks. In this paper, we present a transform domain approach for image deblocking using a deep neural network called DCTResNet. Our application is compressed video motion deblur, where the input video frame has blocking artifacts that make the deblurring task very challenging. Specifically, we use a block-wise Discrete Cosine Transform (DCT) to decompose the image into its low and high-frequency sub-band images and exploit the strong sub-band specific features for more effective deblocking solutions. Since JPEG also uses DCT for image compression, using DCT sub-band images for image deblocking helps to learn the JPEG compression prior to effectively correct the blocking artifacts. Our experimental results show that both PSNR and SSIM for DCTResNet perform more favorably than other state-of-the-art (SOTA) methods, while significantly faster in inference time.**

I. Introduction

JPEG is the most widely used lossy image compression technique. It uses block-wise 2D-DCT to convert images into transform domain and performs compression on each block independently by eliminating and quantizing some of the high-frequency information from the DCT coefficient of the image. However, lossy compression sometimes will also introduce unpleasant distortions in the image called blocking artifacts, which are often seen as a sharp change in intensity at block boundaries. Blocking artifacts can have different magnitude, from mild to severe, depending on the compression strength. Blocking artifacts not only degrade the visual quality of the image but also affect many other image enhancement tasks, as the visually unpleasant blocking artifacts may get enhanced by those image enhancement algorithms, resulting in an even worse visual quality. Therefore, image deblocking can not only help improving the compressed image quality, but also play an important role as the first step of other enhancement tasks to improve their effectiveness on images with blocking artifacts. In this paper we focus on removing the blocking artifacts with experiments conducted on a dataset collected for compressed image motion deblur.

Many image based training dataset for deep learning models are scraped from the internet, such as the ImageNet dataset [1]. The uploaded images are usually compressed to JPEG to reduce their size. Despite the fact that these compressed images preserve the image's global information, the majority of its information is removed that in turn introduces blocking artifacts which can significantly affect the deep learning model performance and result in poor visual quality or lower quantitative results. Therefore, image deblocking can be used as the first stage of processing network for datasets to improve performance in vision tasks.

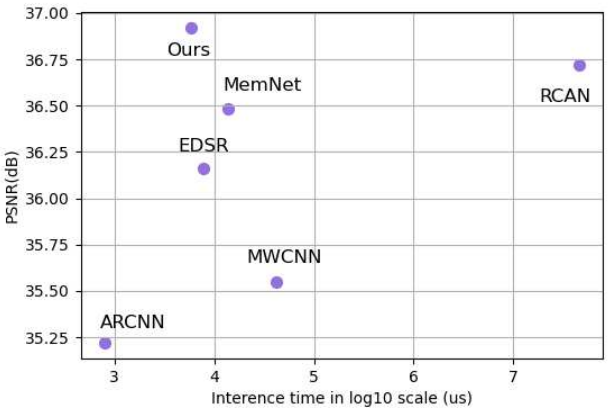

Fig. 1: Plot showing the reconstructed RGB PSNR vs the inference time in log10 scale (microseconds) for RGB image of size 1280x720 of our method compared to other SOTA methods.

Traditional methods like block filtering [2, 3, 4], sparse representation [5] etc. were used for removing the blocking artifacts. However, these methods introduce blurriness in the image and are usually computationally expensive. Deep learning-based methods, such as ARCNN [6], MemNet [7], have recently been used to remove JPEG compression artifacts. These methods have shown promising results for image deblocking. Typically, these methods execute deblocking in the spatial domain (RGB or grayscale channels). ARCNN [6] uses a very shallow model as a deblocking network. It is extremely fast but its performance is limited. It is not able to remove the blocking artifacts from the image when subjected to a lower quality factor JPEG image. MemNet [7] on the other hand uses a memory block, consisting of the recursive unit and gated unit to improve the deblocking performance. But with the increase in depth of network and use of memory blocks, the inference time is increased significantly. In addition, both of these methods are proposed to operate in gray-scale images only and when applied to a color image they do not perform well. Other methods like DnCNN [8], EDSR [9] and RCAN [10] are specifically designed to work on spatial domain and have higher inference time.

A few methods have been developed for processing images in the transform domain for restoration tasks. MWCNN

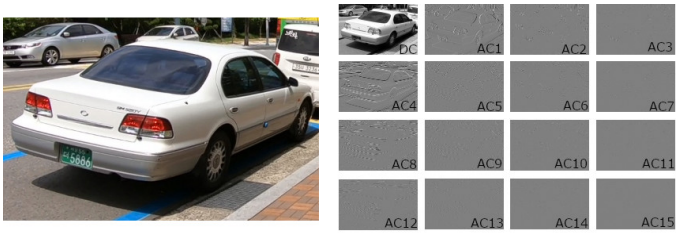

Fig. 2: [a] Input image. [b] Image after applying 4×4 DCT on R channel of input image [a] and then subsampled to 16 sub-band images represents low-frequency DC image and high-frequencies AC1-AC15 images. Sub-band images are $1/4^{th}$ the resolution of input image.

[11, 12] in particular uses Discrete Wavelet Transform (DWT) to decompose the image into its low and high-frequency components to perform image restoration. However, because the entire network is driven by the spatial loss function, the learning still remains in the spatial domain.

In this paper, we present a novel method for deblocking images in the transform domain. We used Discrete Cosine Transform (DCT) as our method to decomposing the image into the transform domain. JPEG compression employs the DCT transform during compression, and the blocking artifact is caused primarily by removing some information from these DCT blocks. Learning in the transform domain allows us to get knowledge of JPEG compression prior which helps to effectively remove blocking artifacts. More importantly, we used 4×4 DCT to decompose the compressed image and subsample the resulting DCT image to create 16 sub-band images, each channel representing DC, AC1, AC2,..., AC15. DC represents the low-frequency component and has global information of the image. AC1 to AC15 represents high-frequency components of the image. AC1 represents the principal vertical component, AC4 representing the principal horizontal component, and AC5 representing the principal diagonal component. Whereas the rest of the ACs represent the subsidiary vertical, horizontal and diagonal components. Fig 2 [b] shows more detail on DCT sub-band images where image [a] is decomposed using 4×4 DCT and subsampled to its respective frequency components.We used a deep residual learning-based network [13] to perform deblocking and reconstruct the image in the transform domain. Finally, we apply the Inverse Discrete Cosine Transform (IDCT) to the reconstructed DCT sub-bands to convert them back to their spatial (RGB) domain. We used REDS motion blur with JPEG compression dataset [14] to train and evaluate the effectiveness of our method for removing blocking artifacts from compressed motion blur images. REDS dataset consist of the synthetic motion blur image, which is created by combining multiple images to simulate the camera and object motion. This motion blur image is then compressed using JPEG with a quality factor of 25, which introduces the JPEG compression artifacts in the input image. Therefore our method of deblocking in the transform domain can be used as a preliminary stage in the image deblurring process.

The main contributions of this paper are as follows:

- Sub-band image-based deblocking: We proposed to use transform domain processing for image deblocking instead of the spatial domain. We applied 4×4 DCT transform as our image decomposition method and used the subsampled DCT sub-band images {DC, AC1, AC2, ... , AC15 sub-images} as input to our network.
- Much faster inference: We conducted extensive experiments on the REDS motion blur with JPEG compression dataset [14] and achieved significant PSNR/SSIM gain over pixel-based deblocking methods, along with a significantly lighter model resulting in faster inference time.

II. Proposed Method

A. Transform Domain

In our work, we used 4×4 DCT transform to decompose the image to its low-frequency and high-frequency component. We subsampled the DCT image to its respective sub-band images as shown in Fig. 2. We then concatenate them together to form 48 channel $\frac{H}{4} \times \frac{W}{4}$ images. Both input and ground truth images are converted using this method.

The advantage of learning in the transform domain for deblocking is in two folds. Firstly, learning in the transform domain for deblocking tasks helps to exploit the information from DCT components by learning the prior knowledge of JPEG compression. It makes the network easier to learn the degradation. Secondly, since the 4×4 DCT sub-band image is $1/4^{th}$ the size of the original image, the effective receptive field is much larger than the previously proposed methods such as EDSR [9], RCAN [10], MemNet [7]. Thus, even with similarly configured network, we were able to train a model that outperforms methods learning in the spatial domain.

B. Network Architecture

Inspired by the success of residual learning architecture in image reconstruction, we modified EDSR to add an additional sub-band specific pixel residue showed by red arrow in Fig. 3 [b]. We also removed the Upsampler block from the EDSR network as input and output of our network are of the same resolution. Our DCT based residual learning network consists of 20 ResBlocks with 64 feature maps. Each ResBlock consists of two convolutional layers sandwiching the ReLU activation function. The structure of the network is shown in Fig. 3. A total of 16 separate networks are trained for learning 16 different DCT sub-band images. The architecture of all the networks are the same.

The input to DCTResNet is 48 channel (16 DCT sub-band images for each of R, G, and B color channel) DCT sub-band image while the output is 3 channel sub-band specific DCT image. The predicted DCT sub-band images are then converted back to the spatial domain using the Inverse Discrete Cosine Transform (IDCT).

III. Experiments and Results

A. Dataset and Data Preparation

In this work, we used REDS dataset [14] for both training and testing purposes. The REDS dataset includes JPEG com-

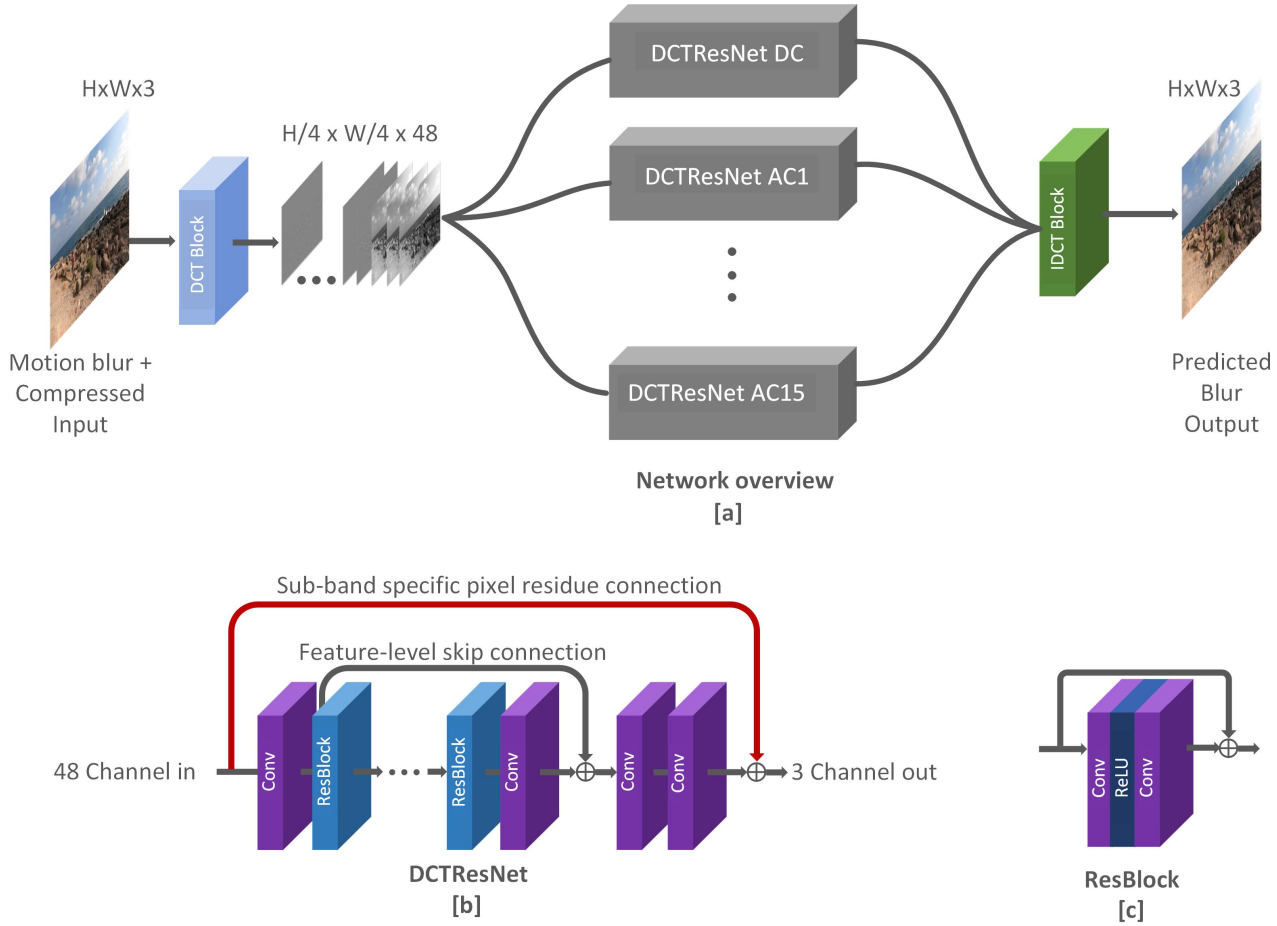

Fig. 3: [a] Network architecture of DCTResNet. RGB image transformed into transform domain using DCT transform. The DCT sub-band images are then used as input to our DCTResNet network. Pixel-level skip connection for learning each corresponding sub-band image. [b] Architecture of DCTResNet. It consist of 20 ResBlocks. [c] Structure of residual block.

pressed motion blur images, its corresponding blur images, and ground truth sharp images. For our purpose of the JPEG deblocking task, we used the first two pairs as our input and ground truth: motion blur JPEG compressed image as input and corresponding motion blur image as ground truth. The JPEG image is compressed using the quality factor of 25.

To prepare the input data for our method, we performed 4×4 block-wise DCT on the input image in each R, G, and B channel. The output of 4×4 DCT is subsampled to its low-frequency (DC) and high-frequency (AC1-AC15) as shown in Fig. 2. We stacked all of the sub-images together to form 48 channel DCT sub-image (16 sub-bands for R, G, and B color channels) which we used as input to our network. As learning is done in the transform domain, the ground truth image is also converted to the transform domain.

While training, we converted all the training data to the transform domain offline and used this newly created DCT dataset to train our network. Creating DCT sub-band images offline helped us to reduce the training time. A total of 24,000 images from 240 different scenes were used for training. For data augmentation, we performed random cropping, horizontal, and vertical flipping of the DCT sub-band images.

While testing, we use padding in the RGB image to align with the 4×4 DCT block if the input is not a perfect factor of 4 which is equivalent to DCT block size. 300 images from 30 different scenes were used to evaluate as mentioned in the REDS dataset website.

B. *Experimental Settings*

TABLE I: Comparison of our proposed DCTResNet with the existing SOTA methods for REDS motion blur with JPEG compression dataset [14]. **Bold** indicates the best result and underline the second best.

Methods	PSNR	SSIM
ARCNN [6]	35.22	0.9344
MWCNN [12]	35.55	0.9263
EDSR [9]	36.16	0.9430
MemNet [7]	36.48	0.9468
RCAN [10]	36.72	0.9476
DCTResNet(ours)	**36.92**	**0.9760**

We trained our model with a single NVIDIA 1080Ti GPU. Model optimization was performed using L_1 loss and Adam optimizer with learning rate initially set to 1×10^{-4} and decreased by a factor of 2 at epochs 100, 150, and 180, with a total of 200 epochs. The patch size was set to 128×128 with each batch containing 16 images.

TABLE II: Comparison of MSE of all the reconstructed DCT subimages of our proposed method with RCAN for REDS motion blur with JPEG compression dataset [14]. **Bold** indicates the best result. All results are in range of 10^{-3}. (Lower value of MSE is better.)

Methods	DC	AC1	AC2	AC3	AC4	AC5	AC6	AC7	AC8	AC9	AC10	AC11	AC12	AC13	AC14	AC15
RCAN	31.449	7.417	**2.886**	**0.752**	7.384	2.704	1.276	**0.350**	3.202	1.471	0.706	0.195	0.988	0.469	0.229	0.069
Our	**31.348**	**7.415**	2.889	0.755	**7.362**	**2.695**	**1.273**	0.350	**3.191**	**1.467**	**0.704**	**0.194**	**0.986**	**0.469**	**0.228**	**0.068**

TABLE III: Table showing inference time DCTResNet compared with the existing SOTA methods for image size of 720×1280 RGB image. **Bold** indicates the best result and underline the second best.

Methods	**Inference time**
ARCNN [6]	**0.8 ms**
MWCNN [12]	41.83 ms
EDSR [9]	7.5 ms
MemNet [7]	13.8 ms
RCAN [10]	45.5 sec
DCTResNet(ours)	5.9 ms

C. Comparison to state-of-the-art methods

For the deblocking task, since the input image we used is motion blurred with JPEG compressed image, we took ground-truth as blur image without any compression. There are no available results as a baseline. Therefore, for a fair comparison, we trained the SOTA methods for image deblocking on the REDS motion blur with JPEG compression dataset [14]. We used ARCNN [6], MemNet [7], MWCNN [12], EDSR [9] and RCAN [10] to evaluate with our method. For all these methods we train the network on REDS motion blur with JPEG compression dataset from scratch based on the parameter provided in their respective papers. Considering the input and output are in RGB color space, all the methods are also trained and tested on color images. Table I shows the comparison of our method with the SOTA. We used Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) [15] as evaluation metrics. From Table I we see that DCTResNet has a 0.2 dB PSNR gain over RCAN that is 10 times deeper than DCTResNet. Compared to EDSR, for same depth of network and similar training configuration, our DCTResNet has 0.76 dB PSNR gain. Also, DCTResNet has significantly higher SSIM value of 0.9760 compare to other methods. (Higher values of PSNR and SSIM are better.)

We further performed a detailed experiment on our proposed method to evaluate the mean squared error (MSE) of each learned DCT sub-band image. We calculate MSE for each of the predicted DCT sub-band images with respect to DCT of ground truth blur image and showed in the Table II. For comparison, we use the best performing method from above Table I and computed its MSE as well. We decompose the reconstructed RGB image from RCAN [10] using the DCT transform and then compute MSE with respect to ground truth image. MSE from our DCT prediction network is higher for the DC component, which contains most of the energy than that of RCAN [10]. While for the ACs, most of the ACs have better MSE results for our method except AC2, AC3, and AC7. The bold text indicated the best results (Lower MSE value is better).

Table. III shows the comparison between our method with SOTA methods in terms of average inference time for processing 720×1280 RGB image on 1080Ti GPU. As our method run in parallel GPU setting the effective average inference time is 5.9 ms per image which is much faster than RCAN that has an average inference time of 45.5 seconds. Even with the cascade configuration of the 16 DCT networks, the total inference time is only 94.4 ms for our DCTResNet. Though ARCNN [6] has the fastest average inference time of around 0.8 ms, the PSNR for ARCNN is 1.7 dB less than that of DCTResNet. Since EDSR [9] is processing in full resolution of the image in spatial domain compared to $\frac{1}{4}^{th}$ of the image resolution for DCTResNet, our method is 1.2 times faster than EDSR for a similar network configuration.

IV. CONCLUSION

We presented DCTResNet, a sub-band image-based deep-learning network that performs image deblocking in a transform domain. We process an image in the transform domain to exploit the information from decomposed low and high-frequencies sub-bands. The reduced spatial size of the DCT sub-band images also improves the receptive field for the convolutional neural network compared to spatial domain processing network with similar networks backbone. From our experiment for the REDS motion blur with JPEG compression dataset, we showed that the learning in transform domain has much better performance of PSNR and SSIM than learning in RGB domain. Even with a smaller network, we got 0.2 dB gain over much deeper network RCAN. Our method with cascade configuration has 480 times faster inference time than RCAN. Additionally, we got 0.76 dB gain over similarly configured spatial domain EDSR network with 1.2 time faster inference time. In the future, we will apply proposed DctRestNet to other low-level vision tasks like image deblurring, denoising, super-resolution, etc. to prove its effectiveness. Finally, we will explore proposed framework for a joint deblur-deblocking solution on real-world captured images.

ACKNOWLEDGMENT

This work is partially supported by the NSF grant 1747751.

REFERENCES

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

[2] P. List, A. Joch, J. Lainema, G. Bjontegaard, and M. Karczewicz. Adaptive deblocking filter. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):614–619, 2003. doi: 10.1109/TCSVT.2003.815175.

[3] Howard C Reeve III and Jae S Lim. Reduction of blocking effects in image coding. *Optical Engineering*, 23(1):230134, 1984.

[4] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. In *IEEE Transactions on Image Processing*, pages 2080–2095, Aug 2007.

[5] Cheolkon Jung, Licheng Jiao, Hongtao Qi, and Tian Sun. Image deblocking via sparse representation. *Signal Processing: Image Communication*, 27(6):663–677, 2012. ISSN 0923-5965.

[6] Ke Yu, Chao Dong, Chen Change Loy, and Xiaoou Tang. Deep convolution networks for compression artifacts reduction. *arXiv preprint arXiv:1608.02778*, 2016.

[7] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *Proceedings of International Conference on Computer Vision*, 2017.

[8] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.

[9] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[10] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, 2018.

[11] Pengju Liu, Hongzhi Zhang, Kai Zhang, Liang Lin, and Wangmeng Zuo. Multi-level wavelet-cnn for image restoration. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[12] Pengju Liu, Hongzhi Zhang, Lian Wei, and Wangmeng Zuo. Multi-level wavelet convolutional neural networks. *IEEE Access*, 7:74973–74985, 2019.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[15] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.