

Evolutionary Algorithm Based Residual Block Search for Compression Artifact Removal

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY-SA 4.0

SUBMISSION DATE / POSTED DATE

11-12-2020 / 16-12-2020

CITATION

Shah, Rishil (2020): Evolutionary Algorithm Based Residual Block Search for Compression Artifact Removal. TechRxiv. Preprint. <https://doi.org/10.36227/techrxiv.13368698.v1>

DOI

[10.36227/techrxiv.13368698.v1](https://doi.org/10.36227/techrxiv.13368698.v1)

Evolutionary Algorithm Based Residual Block Search for Compression Artifact Removal

Rishil Shah

Department of Electrical Engineering

S. V. National Institute of Technology

Surat, India

rishil2008@gmail.com

Abstract—Lossy image compression is ubiquitously used for storage and transmission at lower rates. Among the existing lossy image compression methods, the JPEG standard is the most widely used technique in the multimedia world. Over the years, numerous methods have been proposed to suppress the compression artifacts introduced in JPEG-compressed images. However, all current learning-based methods include deep convolutional neural networks (CNNs) that are manually-designed by researchers. The network design process requires extensive computational resources and expertise. Focusing on this issue, we investigate evolutionary search for finding the optimal residual block based architecture for artifact removal. We first define a residual network structure and its corresponding genotype representation used in the search. Then, we provide details of the evolutionary algorithm and the multi-objective function used to find the optimal residual block architecture. Finally, we present experimental results to indicate the effectiveness of our approach and compare performance with existing artifact removal networks. The proposed approach is scalable and portable to numerous low-level vision tasks.

Index Terms—Evolutionary algorithm, Artifact removal, Residual networks, Image processing

I. INTRODUCTION

In recent years of information explosion, multimedia compression has become pivotal and unavoidable for institutions and companies in order to capitalize on limited storage and transmission capabilities. With image acquisition devices becoming increasingly ubiquitous, the growth rate of image data far exceeds the advancement in compression techniques. The goal of image compression is to minimize redundancies within the image for storage and transmission at low bit rates. Compression techniques are of two types: 1) lossy (e.g. JPEG, JPEG2000) and 2) lossless (e.g. BMP, PNG) [1]. Lossless compression retains the image quality and provides the perfect reconstruction of the original image while lossy compression offers higher compression ratios. Consequently, lossless compression is used for medical images, scientific drawings, and other applications where any divergence from the original image is unacceptable. However, the majority of image data serves purposes where the minor loss of quality for a considerable reduction in bit rate is preferable [1].

Owing to the resounding success of deep convolutional neural networks (CNNs) in high-level computer vision tasks and increased difficulties in enhancing the coding performance

of the traditional hybrid coding based frameworks, end-to-end optimized CNN frameworks for image compression are being extensively developed [2], [14], [15]. In spite of the development of novel compression methods with significantly better performances, JPEG continues to be the most popular lossy image compression standard [16]. With JPEG-compressed images still being widely used in numerous applications ranging from photography to web pages, JPEG’s significance in the present multimedia world is evident.

The JPEG compression standard involves a block-based discrete cosine transform (DCT). A given image is split into 8×8 blocks which are then transformed to frequency domain using 2-D DCT. This is followed by a quantization step, wherein the high frequency information is discarded while retaining low frequency information. The quantization of DCT coefficients introduces ringing and blurring artifacts due to the removal of the high frequency components. In addition to this, the block transform coding approach used in JPEG compression induces blocking artifacts mostly visible in the low-frequency regions [8].

Deep convolutional networks have led to significant improvement in performance for low-level image processing tasks like single image super-resolution [4], [11]–[13] and image denoising [6], [7], [17]. Dong et al. [3] applied the end-to-end convolutional network framework inspired from SRCNN [4] for compression artifact suppression. The substantial improvement in quality achieved by ARCNN [3], has shifted the focus from edge-aware denoising [18], [19] and sparse recovery algorithms [20], [21] to deep learning based methods. The commercial availability and steep drop in prices of graphical processing units (GPUs) coupled with the development of residual and densely connected networks [22], [23] has led to deep CNNs being used for numerous image processing applications. CAS-CNN which uses a 12-layer ConvNet [5] and the L8 architecture with skip connections [8] are widely regarded architectures for compression artifact removal. Such deep CNN based methods used for low-level image processing tasks such as image denoising and artifact removal emphasize on increasing network parameters to enhance performance. Nevertheless, the convolutional layers in such manually designed network usually share the same architecture specifications, which might not always be optimal for the task, affecting the performance and capacity of the

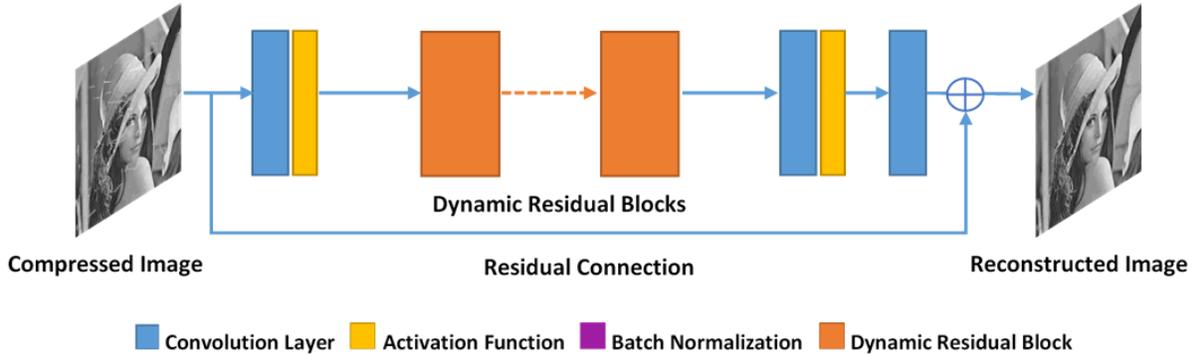


Fig. 1. Dynamic residual network (DRN) architecture with residual learning enabled.

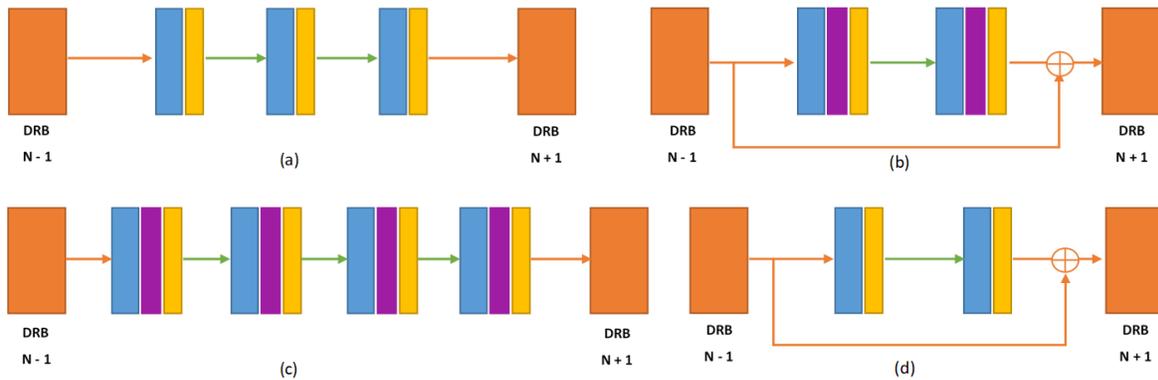


Fig. 2. Examples of dynamic residual block (DRB) architectures included in the search space.

entire network. This leads to sub-optimal image enhancement and inefficient use of computational resources.

In recent times, neural architecture search (NAS) has been successful in achieving state-of-the-art performance on image segmentation [24], denoising [25] and restoration [26] tasks. Although, NAS has been explored recently in the proximate domain of image super-resolution, it has not yet been investigated for compression artifact removal. Given an elastic search space, NAS was able to generate high-quality models with performance comparable to the state-of-the-art in image super-resolution [28]. However, elastic search on the combined micro and macro space requires large amounts of computation and is time-consuming. Instead a more practical approach is outlined by Zhong et al. which focuses on a block-based setup for automatic network generation [27], [29]. The effectiveness of the approach for image super-resolution is substantiated by the performance of the ESRN networks [11].

Given the importance of JPEG images in the present day multimedia world, compression artifact removal is a vastly researched low-level image processing problem. With the advent of deep learning and commercialization of high-end computational resources, numerous CNN based architectures have been proposed. However, all the existing learning based approaches have deep CNN architectures manually-designed by researchers. In this paper, we present an evolutionary-based approach to find an optimal block-based architecture for artifact removal in JPEG-compressed images. We begin by outlining a strategy for developing a deep CNN network

with skip connections through sequential concatenation of dynamic residual blocks (DRBs) found using evolutionary search. Keeping in mind the convergence and scaling issues encountered when training deeper networks for low-level vision tasks [3], [8], [12] we include training hyper-parameters such as residual learning and weight initialization in the search space. We describe an integer-based encoding strategy for the network which allows the direct application of established evolutionary strategies and operations. We use the commonly reported benchmark datasets and performance metrics in our experiments [3], [5], [6], [8], [21], [30].

II. RELATED WORK

A. Compression Artifact Removal

Deep learning based methods have prevailed as the state-of-the-art for nearly all high-level computer vision problems. Following the breakthrough by Dong et al. [4] in achieving remarkable image super-resolution using SRCNN, a similar trend has ensued in the parallel domains of image super-resolution and artifact removal. The compact and efficient artifact removal network - ARCNN [3] improved upon the SRCNN architecture by embedding multiple feature enhancement layers, leading to substantial improvement in image quality compared to traditional artifact removal methods.

To address the convergence issues faced by Dong et al. while training deeper CNN networks [3], Svoboda et al. adopted skip connections, residual learning and symmetric weight initialization in their artifact removal networks [8].

In the same year, Cavigelli et al. presented the CAS-CNN architecture with hierarchical skip connections and the replacement of the conventional mean square error (MSE) loss with the multi-scale loss function [5]. The modifications in the network and training procedure allowed the 12-layer ConvNet to converge beyond state-of-the-art accuracy.

In contrast to developing deeper CNN networks, Guo et al. proposed DDCN, a dual-domain CNN-based model that integrated DCT-domain prior with a CNN architecture [21]. The dual-domain approach was further incorporated in the DMCNN model, which was based on an auto-encoder style architecture and used multi-scale loss [17]. Furthermore, Galteri et al. used a generative adversarial network for artifact removal [30] while the SFCNN architecture [10] provided improvement in performance at reduced computational costs.

B. Deep Residual Networks

Residual networks have achieved substantially better results compared to their shallower CNN counterparts in numerous computer vision problems. In the past few years, they have been widely investigated for image super-resolution [11]–[13], [31]–[33] and compression artifact removal [7]–[9] architectures. Lim et al. identified the short-comings of directly implementing the ResNet architecture [22] for low-level vision problems, and instead suggested a modified residual block for optimal super-resolution performance [31]. Their enhanced deep residual network – EDSR improved upon the existing SRResNet [33] by removing the computationally expensive batch normalization layers and adopting residual scaling.

However, all the preceding deep residual networks failed to incorporate hierarchical features for reconstruction. Zhang et al. addressed this flaw by introducing a residual dense network (RDN) made up of smaller residual dense blocks (RDBs) [13]. The RDN network has a contiguous memory mechanism incorporated via direct links between preceding RDBs and all layers of the current RDB. To preserve hierarchical features globally the network uses dense feature fusion (DFF) after the extraction of local dense features. Recently, Song et al. proposed the ESRN architecture for image super-resolution achieved by the search for efficient RDBs [11].

III. PROPOSED APPROACH

Defining a proper search space is crucial to reduce the computational costs and enhance the performance of the NAS algorithm. The effectiveness of block based architectures for low-level vision tasks has already been established in existing literature [6], [7], [9], [11], [13], [29], [31]. We first present a dynamic residual network and its corresponding genotype representation using integer values in the following section. Next, we outline the evolutionary search algorithm and operators used.

A. Dynamic Residual Network

The framework for end-to-end CNN for artifact removal consists of four operations: feature extraction, mapping, enhancement and reconstruction [3]. The dynamic residual network (DRN) as shown in Fig. 1, consists of pre-defined

TABLE I
PARAMETER SEARCH SPACE FOR DRB ARCHITECTURE AND NETWORK TRAINING.

Block parameters	Parameter values	Allele
Block state	{Not active, Active}	{0, 1}
Convolutional layers	{1, 2, 3, 4}	{0, 1, 2, 3}
Number of filters	{8, 16, 32, 64}	{0, 1, 2, 3}
Filter size	{1, 3, 5, 7, 9, 11}	{0, 1, 2, 3, 4, 5}
Batch normalization	Absent, Present	{0, 1}
Activation function	{Leaky ReLU, ReLU, SELU}	{0, 1, 2}
Skip connection	Absent, Present	{0, 1}
Training parameters	Parameter values	Allele
Residual learning	Absent, Present	{0, 1}
Weight initialization	{None, Gaussian, Symmetric}	{0, 1, 2}

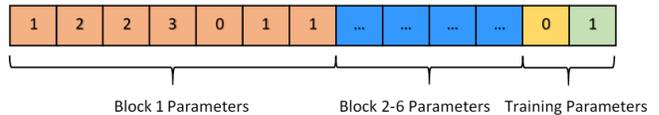


Fig. 3. Example of a DRN genotype.

feature extraction and reconstruction layers based on the L8 architecture [8]. Our work focusses on finding the more important feature enhancement and non-linear mapping layers using evolutionary search of the intermediate dynamic residual blocks (DRBs). The DRB architecture given in Fig. 2(b) utilizes skip connections and batch normalization layers in addition to the convolutional layers. Each DRN is mapped to a chromosome of length 44, where the first 42 genes model 6 DRBs with 7 genes each and the last 2 genes represent residual learning and weight initialization respectively. The chromosome length is limited to 44 to reduce the computational cost of the evolutionary search and can be increased as per the availability of resources. Each DRB is encoded as {block state, block parameters} where the value of the block state decides whether the block is included in the network or not. Each parameter is mapped in the genotype by using a 0-based integer encoding strategy (Table 1) which allows for the direct use of established evolutionary operators. Fig. 3 shows a sample chromosome with 1st DRB and training parameters mapped as shown in Table 1. The value 1 of the first gene indicates that the 1st DRB is included in the final network.

B. Evolutionary algorithm based Residual Block Search

1) *Evolutionary Algorithm*: The evolutionary algorithm used for architecture search is outlined below. Firstly, an initial population of size P is randomly chosen and the fitness of each individual is evaluated. Now, we set aside the elite individuals E_m i.e. the fittest individuals, based on the pre-defined elitism number T . The elitist approach preserves the fittest individuals as we evolve the populations assuring that the quality of the candidate solutions does not degrade as we evolve from one generation to another. At the start of each generation cycle we choose the *parents* from P using tournament selection. The *offsprings* are generated using the simulated binary crossover of the selected *parents* based on the given crossover probability p_{cross} . Next, the mutation of the *offsprings* is

carried out by the bounded polynomial mutation operator producing the *mutated* individuals. A high mutation probability p_{mut} promotes exploration of new candidate solutions but at the same time prevents exploitation of fit individuals within the current generation. After mutation, we evaluate the fitness of the newly generated individuals and update P and E_m accordingly. The evolutionary search proceeds as follows:

Input: population size P , maximum number of generations M , crossover probability p_{cross} , mutation probability p_{mut} , elitism number T , training dataset D_{train} , validation dataset D_{val} .

Initialize: (i) fitness F_P of all individuals
(ii) elitism $E_m \leftarrow Update(F_P, T)$

while $m \leq M$ **do**

$parents \leftarrow Selection(P, F_P, T)$
 $offsprings \leftarrow Crossover(parents, m_{cross})$
 $mutated \leftarrow Mutation(offsprings, m_{mut})$
 $models \leftarrow Train(mutated, D_{train})$
 $F_P \leftarrow Evaluate(models, D_{val})$
 $E_m \leftarrow Update(F_P, T, E_{m-1})$
 $P \leftarrow mutated + E_m$
 $m = m + 1$

end while

Output: elitism E_m

2) *Fitness Evaluation:* For determining the fitness of each individual solution we first map the genotype to its corresponding DRN structure and training parameters. Then we train the network on the D_{train} dataset for a given number of epochs and evaluate its performance on the validation dataset D_{val} . For compression artifact removal, the peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) are the most widely used performance metrics [3], [5], [6], [8], [21], [30]. Hence, during the evolution we use both the metrics to evaluate the performance of the candidate solutions. In order to explore efficient networks we also use the number of parameters as an objective in one of the evolutionary search trials. Hence, we consider three objectives for the artifact removal task,

- quantitative metric to evaluate noise removal capability of the network (PSNR),
- quantitative metric to evaluate structural content retrieval capability of network (SSIM),
- number of network parameters.

Overall, this translates to a multi-objective problem for which we use the NSGA-II method [28].

IV. EXPERIMENTS

A. Implementation

We use the BSD500 [34], LIVE1 [35] and Classic5 datasets in our experiments. The BSD500 dataset consists of 200 train and 200 validation images which are merged to obtain the training dataset while the 29 LIVE1 and 5 Classic5 images are used as the test dataset. The RGB images are transformed using the YCbCr color model and only luminance

TABLE II
EVOLUTIONARY SEARCH PARAMETERS AND FITNESS EVALUATION OBJECTIVES.

Search parameters	1 st search	2 nd search	3 rd search
Maximum generations	4	6	6
Population size	30	25	30
Mutation probability	0.2	0.2	0.3
Elitism number	5	5	5
Number of epochs	20	20	10
Tournament size	2	2	3
Evaluation objective	1 st search	2 nd search	3 rd search
PSNR	Yes	Yes	Yes
SSIM	Yes	Yes	Yes
Network parameters	No	No	Yes

TABLE III
EVOLUTIONARY SEARCH BEST PERFORMING MODELS.

Performance metric	DRN-Es1	DRN-Es2	DRN-Es3
PSNR	28.26	28.09	27.95
SSIM	0.794	0.791	0.785
Network parameters	533k	495k	313k

- Y component is retained. In accordance with previous papers, the training as well as test images are compressed using the MATLAB JPEG encoder with a quality factor (QF) of 10. The experiments are carried out using the PyTorch framework [37] for deep learning and the DEAP framework [36] for evolutionary search, on a server with 12GB Tesla K80 GPU.

B. Searching Detail

For the training of individual networks during the evolutionary search we use 32×32 randomly cropped patches from the BSD500 training set images. The 400 images are augmented using scaling and rotation to increase the data size. A mini-batch size of 64 is used and the L1-norm is used as the loss criterion between the original and the reconstructed images. We use train images of size 32×32 during the evolutionary search to accommodate a much larger training set within reasonable computation time. After training for the given number of epochs, each candidate solution is evaluated on the 29 LIVE1 dataset images. As mentioned in Table 2, the 1st and 2nd evolutionary search experiments only evaluate the models on the basis of the reconstruction quality, whereas the 3rd experiment also takes into account the network parameters. We increase the maximum generations in our experiments to emphasize on exploration over exploitation. The performance and number of parameters of the best network produced in each search is given in Table 3. The best network generated in the n th evolutionary search is referred to as DRN-Esn.

C. Retraining Detail

After the evolution, the best performing architectures are retrained using 64×64 randomly cropped patches from the training dataset. A mini-batch size of 64 is used and similar to the search stage, the L1-norm is used as the loss criterion. We initialize the Adam optimizer with a learning rate of 10^{-3}

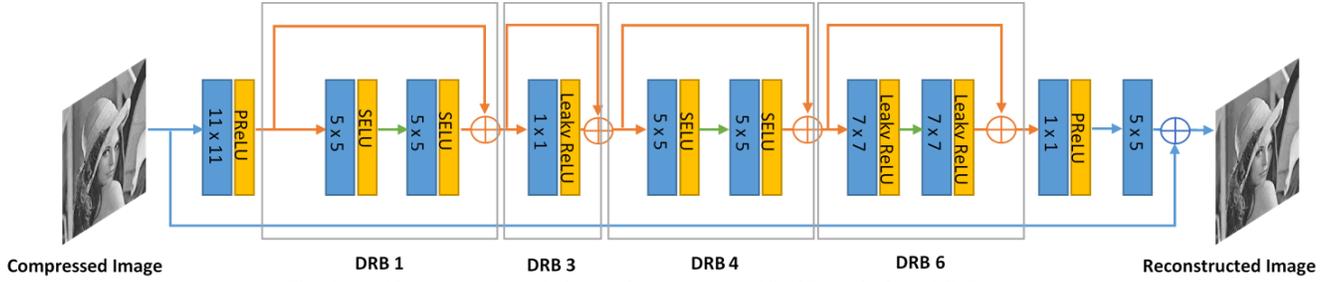


Fig. 4. Artifact removal result for the image “monarch” (QF = 10) from LIVE1 dataset.

TABLE IV
RECONSTRUCTED IMAGE QUALITY COMPARISON FOR LIVE1 AND CLASSIC5 DATASETS.

Dataset	Quality	Metric	JPEG	SA-DCT [18]	ARCNN [3]	Fast-ARCNN [3]	L4 [8]	DnCNN-3 [6]	DRN-Es1 (ours)	DRN-Es2 (ours)	DRN-Es3 (ours)
LIVE	10	PSNR(dB)	27.77	28.65	29.13	29.10	29.08	29.19	29.25	29.22	29.18
		SSIM	0.791	0.809	0.823	0.824	0.824	0.826	0.831	0.826	0.826
Classic5	10	PSNR(dB)	27.82	28.88	29.04	29.03	-	29.40	29.34	29.32	29.31
		SSIM	0.780	0.807	0.811	0.812	-	0.820	0.822	0.819	0.818

which is scaled down by 2 for every 20k iterations. We set parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The number of iterations is fixed at 100k which is 100 times less than 10^7 used to train ARCNN [3], and about 0.5 times the 250k iterations reported in [8].

D. Results and Discussions

The comparison of the performances of the searched architectures with the reference artifact removal models is given in Table 4. The PSNR of 29.25 dB and SSIM of 0.831 achieved by DRN-Es1 is a significant improvement over the ordinary JPEG-compressed images. As indicated the performance of all 3 generated networks is at par with other CNN-based artifact removal networks. Based on the evolutionary search results, we were able to infer that residual learning provides substantial improvement in performance as well as accommodates a higher training rate. Our attempts to increase the learning rate up to 0.1 using adjustable gradient clipping [8] proved unfruitful, with subpar model performances that we choose not to include.

Another important observation based on the search results, was the absence of the batch normalization layers in the majority of elite networks. This was particularly evident in the case of the 3rd evolutionary search, which shows that batch normalization layers provide little quality improvement but increase network parameters considerably.

The limited number of training epochs during the evolutionary search restricts search, reducing computation time as well as generating networks that converge quickly. The DRN-Es1 model is able to surpass the performance of ARCNN within just 40k iterations. In spite of the smaller network size, the DRN-Es3 shows reasonably good improvement in image quality surpassing all other methods except DnCNN-3 [6].

However, we reason that the performances of the generated networks are limited by the relatively small number of iter-

ations used for re-training. The smaller dataset and training iterations reduces the computational cost but at the same time limits the improvement in image quality. For the DRN-Es1 in particular, a much larger dataset with increased image size should provide substantial improvement in reconstructed image quality. Fig. 5 shows the reconstructed images by all three networks.

V. CONCLUSIONS AND FUTURE WORKS

In this paper, we have presented an evolutionary algorithm based search for compression artifact removal. Most existing artifact removal networks are manually-designed by researchers and require domain expertise. On the other hand, the proposed evolutionary search is able to generate deep neural networks that overcome the performance limitations of manually-designed networks without necessarily relying on domain knowledge. We propose a residual network architecture, based on residual block components, which is searched from the given parameter space. The generated networks are able to obtain better results than existing CNN-based artifact removal networks with significantly less number of training iterations. With dual-domain networks that integrate spatial and DCT domain information, becoming increasingly popular, we believe the next step of research would be employing evolutionary search to find optimal dual-domain networks. We believe our work provides insights for employing evolutionary algorithm and developing better-suited deep residual networks for low-level vision tasks.

REFERENCES

- [1] Z. Wang, A. C. Bovik, and L. Lu, “Why is image quality assessment so difficult?,” IEEE International Conference on Acoustics Speech and Signal Processing, May 2002.
- [2] S. Ma, X. Zhang, C. Jia, Z. Zhao, S. Wang, and S. Wang, “Image and Video Compression With Neural Networks: A Review,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 30, no. 6, pp. 1683–1698, Jun. 2020.

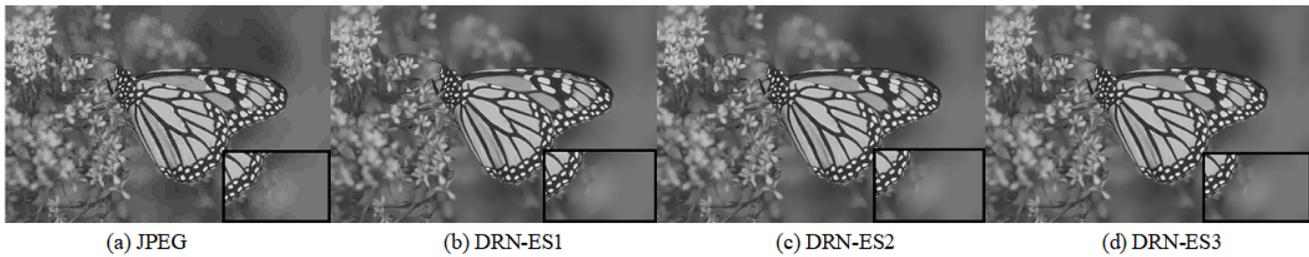


Fig. 5. Artifact removal result for the image “monarch” (QF = 10) from LIVE1 dataset.

- [3] C. Dong, Y. Deng, C. C. Loy, and X. Tang, “Compression Artifacts Reduction by a Deep Convolutional Network,” 2015 IEEE International Conference on Computer Vision (ICCV), Dec. 2015.
- [4] C. Dong, C. C. Loy, K. He, and X. Tang, “Image Super-Resolution Using Deep Convolutional Networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, Feb. 2016.
- [5] L. Cavigelli, P. Hager, and L. Benini, “CAS-CNN: A deep convolutional neural network for image compression artifact suppression,” 2017 International Joint Conference on Neural Networks (IJCNN), May 2017.
- [6] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, “Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising,” *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.
- [7] Y. Tai, J. Yang, X. Liu, and C. Xu, “MemNet: A Persistent Memory Network for Image Restoration,” 2017 IEEE International Conference on Computer Vision (ICCV), Oct. 2017.
- [8] P. Svoboda, M. Hradis, D. Barina, and P. Zemcik: “Compression Artifacts Removal Using Convolutional Neural Networks,” *J. WSCG*, vol. 24, no. 2, pp. 63–72, 2016.
- [9] B. Zheng, R. Sun, X. Tian, and Y. Chen: “S-Net: a scalable convolutional neural network for JPEG compression artifact reduction,” *Journal of Electronic Imaging*, vol. 27, no. 4, p. 043037, 2018.
- [10] T. Kim, H. Lee, H. Son, and S. Lee, “SF-CNN: A Fast Compression Artifacts Removal via Spatial-To-Frequency Convolutional Neural Networks,” 2019 IEEE International Conference on Image Processing (ICIP), Sep. 2019.
- [11] D. Song, C. Xu, X. Jia, Y. Chen, C. Xu, and Y. Wang, “Efficient Residual Dense Block Search for Image Super-Resolution,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, pp. 12007–12014, Apr. 2020.
- [12] J. Kim, J. K. Lee, and K. M. Lee, “Accurate Image Super-Resolution Using Very Deep Convolutional Networks,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2016.
- [13] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, “Residual Dense Network for Image Super-Resolution,” 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Jun. 2018.
- [14] J. Balle, V. Laparra, and E. P. Simoncelli, “End-to-end optimized image compression,” arXiv preprint arXiv: 1611.01704, 2016.
- [15] L. Theis, W. Shi, A. Cunningham, and F. Huszar, “Lossy image compression with compressive autoencoders,” arXiv preprint arXiv: 1703.00395, 2017.
- [16] G. Hudson, A. Léger, B. Niss, I. Sebestyén, and J. Vaaben, “JPEG-1 standard 25 years: past, present, and future reasons for a success,” *Journal of Electronic Imaging*, vol. 27, no. 04, p. 1, Aug. 2018.
- [17] X. Zhang, W. Yang, Y. Hu, and J. Liu, “Dmccn: Dual-Domain Multi-Scale Convolutional Neural Network for Compression Artifacts Removal,” 2018 25th IEEE International Conference on Image Processing (ICIP), Oct. 2018.
- [18] 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, May 2007.
- [19] 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, Aug. 2007.
- [20] X. Liu, X. Wu, J. Zhou, and D. Zhao, “Data-driven sparsity-based restoration of JPEG-compressed images in dual transform-pixel domain,” 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2015.
- [21] J. Guo and H. Chao, “Building Dual-Domain Representations for Compression Artifacts Reduction,” *Lecture Notes in Computer Science*, pp. 628–644, 2016.
- [22] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2016.
- [23] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jul. 2017.
- [24] C. Liu, L.-C. Chen, F. Schroff, H. Adam, W. Hua, A. L. Yuille, and L. Fei-Fei, “Auto-DeepLab: Hierarchical Neural Architecture Search for Semantic Image Segmentation,” 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2019.
- [25] H. Zhang, Y. Li, H. Chen, and C. Shen, “Memory-Efficient Hierarchical Neural Architecture Search for Image Denoising,” 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2020.
- [26] H. Zhang, Y. Li, H. Chen, C. Shen: “Ir-nas: Neural architecture search for image restoration,” arXiv preprint arXiv: 1909.08228.
- [27] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter: “Neural architecture search: A survey,” arXiv preprint arXiv: 1808.05377, 2018b.
- [28] Chu X., Zhang B., Ma H., Xu R., Li J. and Li Q.: “Fast, accurate and lightweight super-resolution with neural architecture search,” arXiv preprint arXiv: 1901.07261.
- [29] Z. Zhong, J. Yan, W. Wu, J. Shao, and C.-L. Liu, “Practical Block-Wise Neural Network Architecture Generation,” 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Jun. 2018.
- [30] L. Galteri, L. Seidenari, M. Bertini, and A. D. Bimbo, “Deep Generative Adversarial Compression Artifact Removal,” 2017 IEEE International Conference on Computer Vision (ICCV), Oct. 2017.
- [31] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, “Enhanced Deep Residual Networks for Single Image Super-Resolution,” 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Jul. 2017.
- [32] J. Kim, J. K. Lee, and K. M. Lee, “Deeply-Recursive Convolutional Network for Image Super-Resolution,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2016.
- [33] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network,” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jul. 2017.
- [34] P. Arbelaez, M. Maire, C. Fowlkes and J. Malik.: “Contour detection and hierarchical image segmentation.” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(5): 898–916, May 2011.
- [35] L. Cormack H.R. Sheikh, Z.Wang and A.C. Bovik, “Live image quality assessment database release 2,” 2014.
- [36] Fortin, F.A., De Rainville, F.M., Gardner, M.A., Parizeau, M., Gagné, C.: “DEAP: Evolutionary algorithms made easy.” *J. Mach. Learn. Res.* 13, 2171–2175 (2012).
- [37] A. Paszke et al.: “Pytorch: An imperative style, high-performance deep learning library,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8026–8037.