Training compression artifacts reduction network with domain adaptation

Ham, Yu-Jin, Yoo, Chaehwa, Kang, Je-Won

Yu-Jin Ham, Chaehwa Yoo, Je-Won Kang, "Training compression artifacts reduction network with domain adaptation," Proc. SPIE 11842, Applications of Digital Image Processing XLIV, 118420U (1 August 2021); doi: 10.1117/12.2597094

SPIE.

Event: SPIE Optical Engineering + Applications, 2021, San Diego, California, United States

Training Compression Artifacts Reduction Network with Domain Adaptation

Yu-Jin Ham[a], Chaehwa Yoo[a,b], and Je-Won Kang[a,b]

[a]Department of Electronic and Electrical Engineering, Ewha W. University, Seoul, South Korea
[b]Graduate Program in Smart Factory, Ewha W. University, Seoul, South Korea

ABSTRACT

Compression artifact removal is imperative for more visually pleasing contents after image and video compression. Recent works on compression artifact reduction network (CARN) assume that the same or similar quality of images would be employed for both training and testing, and, accordingly, a model needs a quality factor as a prior to accomplish the task successfully. However, the possible discrepancy will degrade performance substantially in a target if the model confronts a different level of distortion from the training phase. To solve the problem, we propose a novel training scheme of CARN to take an advantage of domain adaptation (DA). Specifically, we assign an image encoded with a different quality factor as a different domain and train a CARN using DA to perform robustly in another domain of a different level of distortion. Experimental results demonstrate that the proposed method achieves superior performance on DIV2K, BSD68, and Set12.

Keywords: Domain Adaptation, Compression Artifact Reduction, Deep Learning, Video/Image Coding

1. INTRODUCTION

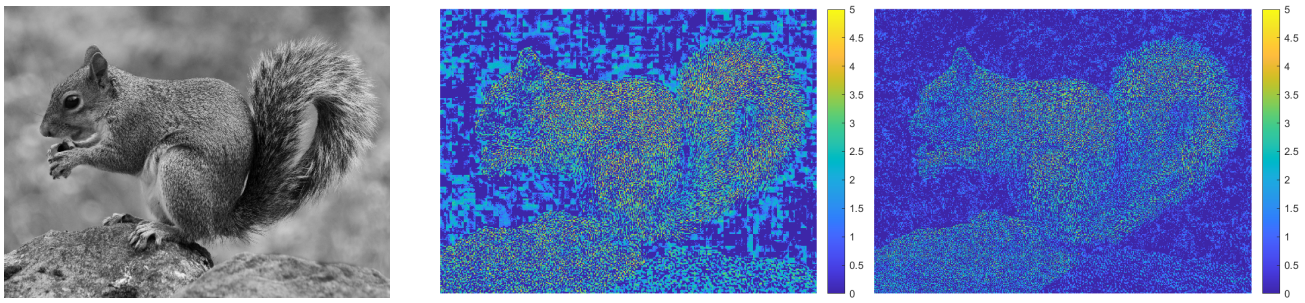
Nowadays, visual content is ubiquitous in our daily life. Image and video dominate internet streaming services and social media, and content providers are actively working to produce high-quality footage. However, while the original content is created and edited without visual artifacts in the production phase, undesired compression artifacts such as blocky and ringing artifacts are unavoidable after compression and transmission. In this regard, in recent years, image and video enhancement techniques have been widely used for quality improvement to obtain more visually pleasing results through post-processing.[1–6]

Compression artifacts reduction network (CARN) is an enabling technique based on a deep neural network (DNN) to reduce the unwanted artifacts after compression.[1–5] In the previous studies, the CARNs were trained under the assumption that similar quality of images was used during both training and testing. However, the performance of a pre-trained model was substantially degraded in a target application when the model needed to handle a different range of distortion or quality factors (QFs) which was unseen during training. Unfortunately, it is usually unknown how different the qualities between the training samples and testing samples are during the post-processing, and such the discrepancies incur the loss. Hence, quality-adaptive artifact reduction is required.

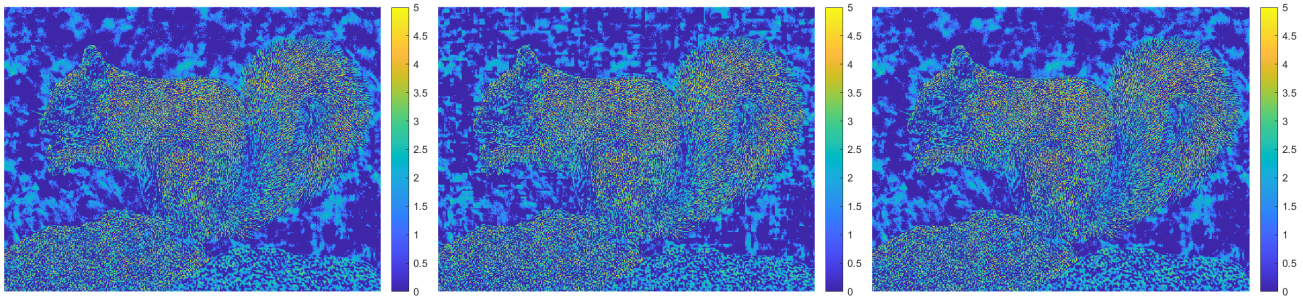
Fig. 1 shows our motivation in which a CARN model differently conducts quantization noise reduction in different training QFs. The residual images are obtained by subtracting the reconstructed images from the ground-truth. Fig. 1b displays the compression error of the JPEG compressed image with $QF = 10$. Fig. 1d, and 1e display the residual images when the model is trained with $QF = 10$ and 70, respectively. It is clearly observed that the model provides degraded performance when the testing QF is different with the training QF. In a naïve approach to solve the problem, a model may use a large training set of image samples including all the possible QFs as shown in Fig. 1f. However, it is challenging for a model to adapt QF of interest during training, and the training time increases dramatically. In contrast, Fig. 1c is the residual of the image reconstructed by our method, which shows the most visually satisfactory result and has the highest numerical gain. Although the training and testing QFs are different, the proposed method alleviates blocky and ringing artifacts from Fig. 1b without compromising brightness change information of the background.

Corresponding author: Je-Won Kang (E-mail: jewonk@ewha.ac.kr)

Applications of Digital Image Processing XLIV, edited by Andrew G. Tescher, Touradj Ebrahimi, Proc. of SPIE
Vol. 11842, 118420U · © 2021 SPIE · CCC code: 0277-786X/21/$21 · doi: 10.1117/12.2597094



(a) Original (b) JPEG ($QF = 10$): 23.7dB (c) Ours (Train $QF = 70$): 24.1dB



(d) CARN (Train $QF = 10$): 24.1dB (e) CARN (Train $QF = 70$): 23.9dB (f) CARN (Train $QF = 10, 70$): 24.1dB

Figure 1: Visualization of the residuals between the reconstructed images of all scenarios and the ground-truth image. (Tested on $QF = 10$) For better visualization, we take logarithm of the absolute value of the residuals added by 1. $log(\mid I_{recon} - I_{gt} \mid +1)$. We present a quantitative result in a format of "method: PSNR (dB)" for (b)-(f). We use RDN as CARN backbone. Best viewed in a digital monitor.

In this paper, we propose a novel training framework of a CARN using domain adaptation (DA) to fill the gap between training and testing QFs. To the best of authors' knowledge, this is the first study to apply the DA to a CARN for adapting a different level of distortion. In our new perspective, images encoded with different QFs are considered as different domains. More precisely, we take advantage of DA by transferring knowledge from source to target,[7–9] so that a feature extractor generates more domain-invariant features. Hence, by applying the DA to our training scheme, we enable a CARN to efficiently handle unseen quality of images in a target application, although feature extractor and denoiser of the network has been pre-trained on the different distortion level ahead. The proposed method does not require ground-truth to the target image samples but a domain label to indicate whether a sample is generated from the same QF or not. Hence, the denoiser of the proposed network is trained in an unsupervised manner on the target side.

2. RELATED WORKS

2.1 Compression Artifact Reduction Networks (CARN)

The CARNs are actively used for post-processing[1–5] and in-loop filters.[10–12] Recently, there have been several studies to develop CARNs. Motivated by deep convolutional neural network (CNN) for image super-resolution,[13] Dong et al. proposed an artifact reduction convolutional neural network (ARCNN) to reduce compression artifacts.[1] Later, various CARN architectures have been developed both on a pixel domain and a transform domain such as a sinogram,[14] a DCT domain,[15] and multiple domains.[16] Recently, there have been several studies on quality adaptive CARNs. Kim et al.[17] proposed a quality-adaptive artifacts removal network based on a gating scheme. Ehrlich M. et al.[18] used a quantization matrix of JPEG to adapt model parameters to a different level of distortion. Although these studies could reduce a wider range of distortion using a single model, they needed to estimate quality information in both training and testing, which challenges end-to-end learning. In contrast, our method does not require any quality information as a prior during testing.

2.2 Domain Adaptation (DA)

The DA is originally developed to shift a feature space from source to target by reducing the discrepancy between two domains – source and target. There have been conventional DA studies such as Maximum Mean Discrepancy (MMD)[7] to calculate a distance between two data distributions and produce features using a kernel-based statistical metric. Researchers have actively conducted DNN-based DA studies in past years.[8,9,19–22] Ganin et al.[8] introduced a domain classifier to discriminate a source and a target domain and devised Gradient Reversal Layer (GRL) to train the network. Adversarial learning was used with a standard backpropagation method. Several works attempted to challenge the adaptation when there are only a few available samples in the target.[9,19] The studies are applied to various computer vision and image processing techniques such as image classification,[9] face recognition,[19] semantic segmentation,[20] and multi-task.[21] However, there are only few works to exploit DA for image enhancement and denoising. In Ref. 22, real-world depth data was enhanced by plentiful synthetic data using DA, alleviating the domain discrepancy between synthetic and real data. However, it focuses on adjusting from the synthetic source domain to real-world target domain due to the lack of a real dataset in the target domain. This work introduces DA into the denoising task of Time-of-Flight depth denoising but still does not address noise levels within the data.

3. PROPOSED METHOD

3.1 Problem Formulation and Objective

We focus on reducing compression artifacts using a DA method so that a single CARN model can handle different qualities of images and provide robust performance for enhancement. We present a training method to tune a CARN to be efficiently applied to test images with a different distortion level from the original training samples. In our problem formulation, we assign two different distortion levels to source and target domains. For instance, a high-quality (HQ) image sample as target domain goes through a CARN that has been trained using a set of low-quality (LQ) images as the source domain, and vice versa.

Our CARN training framework consists of three components: a feature extractor $\mathcal{F}$, a denoiser $\mathcal{D}$, and a quality discriminator $\mathcal{Q}$. The $\mathcal{F}$ extracts features from the source or target input images, while source and target share weights through one common $\mathcal{F}$. The $\mathcal{D}$ is learned by receiving features extracted only from the source images. At this point, the target is not directly involved in training $\mathcal{D}$, and, therefore, the denoiser on the target side is trained in an unsupervised manner Meanwhile, the $\mathcal{Q}$ is aimed at fooling $\mathcal{F}$. The $\mathcal{F}$ is initialized with a pre-trained weight that extracts the features for the source images. As $\mathcal{Q}$ is added to the network, $\mathcal{F}$ and $\mathcal{Q}$ play a minimax game on discriminating source and target and extracting features from $\mathcal{F}$ in order for $\mathcal{D}$ to function properly. The inputs of the network x_i can be both source quality images $x_s \in \mathcal{S}$ and target counterparts $x_t \in \mathcal{T}$. It is noteworthy that $\mathcal{D}$ is trained by images with the source quality only. The corresponding ground-truth of input source images is denoted by y_s, and quality labels which are source or target are represented by q_i $(i = 1, 2)$. Our goal is to adaptively remove compression artifacts on images with a distortion level corresponding to the quality of the target domain in the inference phase where $\mathcal{Q}$ is excluded, and our objective function is as follows:

$$\min_{\theta_{\mathcal{F}},\theta_{\mathcal{D}}} \max_{\theta_{\mathcal{Q}}} L_{\mathcal{D},s} \; - \; \lambda \cdot L_{\mathcal{Q}}, \tag{1}$$

which will be further explained in parameter optimization.

3.2 Network Architecture and Loss Function

We opt to use the state-of-the-art image restoration model Residual Dense Network[2] (RDN) as backbone. The backbone is split into feature extractor and denoiser for our methods. Specifically, the first 2 convolutional layers of RDN and the rest are used as feature extractor and denoiser, respectively. Then, quality discriminator is connected with the feature extractor. It is composed of 3 fully connected layers, and two of which are followed by rectified linear unit (ReLU). We used RDN as a backbone, however, the $\mathcal{D}$ can be chosen from arbitrary CARNs and they are also used for inference. In contrast, the quality discriminator $\mathcal{Q}$ is included during training to conduct DA for which the $\mathcal{F}$ and the $\mathcal{Q}$ play a minimax game on the discrimination loss. In this manner, even though a CARN is pre-trained using images coded with a different quality factor, the model can be adapted to

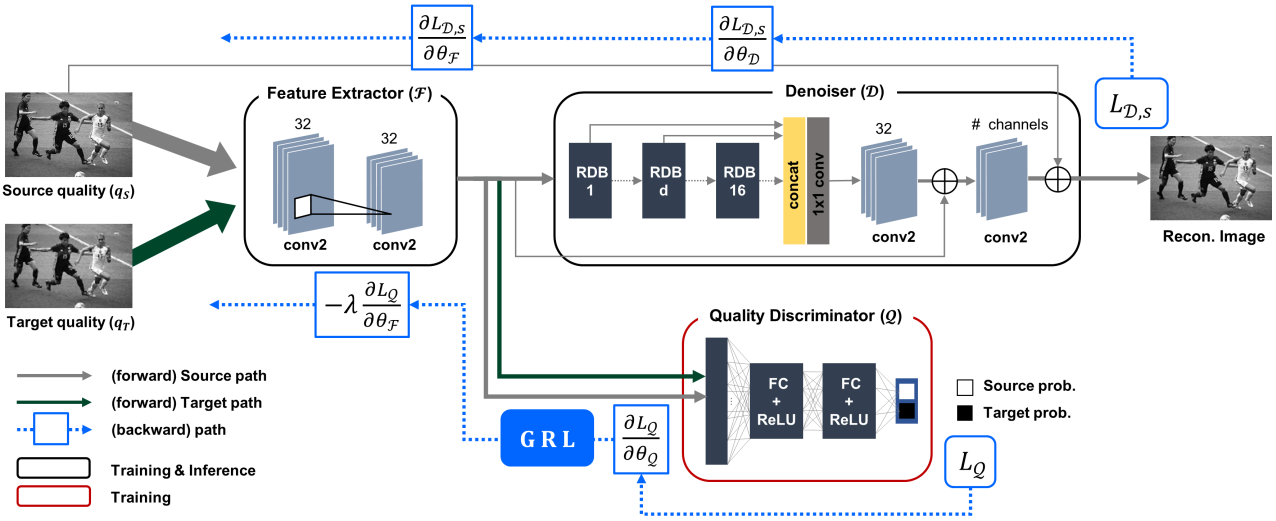


Figure 2: An overview of the proposed network training scheme for compression noise removal.

different level of distortion in a target application. We use two loss terms for training, which are denoising loss $L_{\mathcal{D},s}$ and quality-discrimination loss $L_{\mathcal{Q}}$ as follow:

$$L_{\mathcal{D},s} = \frac{1}{N_s} \sum_{h=1}^{H} \sum_{w=1}^{W} \sum_{c=1}^{C} \left| y_s^{(h,w,c)} - \mathcal{D}\left(\mathcal{F}\left(x_s^{(h,w,c)}\right)\right) \right|, \tag{2}$$

$$L_{\mathcal{Q}} = \frac{1}{N_s + N_t} \sum_{x_i \in \mathcal{S} \cup \mathcal{T}} \sum_{i=1}^{2} q_i \log \mathcal{Q}(\mathcal{F}(x_i)), \tag{3}$$

where input images are in size $H \times W \times C$ and N_s and N_t are the number of source and target quality images, respectively. The denoising loss $L_{\mathcal{D},s}$ is defined as a pixel-level difference between the original image and a distorted image in the source domain only. The quality-discrimination loss $L_{\mathcal{Q}}$ is defined as a cross-entropy loss to predict the quality of an image. We use the conventional stochastic gradient descent (SGD) method while the quality-discrimination loss is reversed using a gradient reversal layer.[8] In summary, the overall loss is defined as in Equation (4):

$$L = \min_{\theta_{\mathcal{F}}, \theta_{\mathcal{D}}} \max_{\theta_{\mathcal{Q}}} L_{\mathcal{D},s} \; - \; \lambda \cdot L_{\mathcal{Q}} \tag{4}$$

where λ is a positive hyper-parameter that adjusts the trade-off between the denoising loss and the quality-discrimination loss. $\theta_{\mathcal{F}}$, $\theta_{\mathcal{D}}$ and $\theta_{\mathcal{Q}}$ are parameters of $\mathcal{F}$, $\mathcal{D}$, and $\mathcal{Q}$, respectively. These parameters are updated as follows:

$$\theta_{\mathcal{D}} \leftarrow \theta_{\mathcal{D}} - \eta \cdot \frac{\partial L_{\mathcal{D},s}}{\partial \theta_{\mathcal{D}}}, \tag{5}$$

$$\theta_{\mathcal{Q}} \leftarrow \theta_{\mathcal{Q}} - \eta \cdot \lambda \cdot \frac{\partial L_{\mathcal{Q}}}{\partial \theta_{\mathcal{Q}}}, \tag{6}$$

$$\theta_{\mathcal{F}} \leftarrow \theta_{\mathcal{F}} - \eta \left(\frac{\partial L_{\mathcal{D},s}}{\partial \theta_{\mathcal{D}}} \cdot \frac{\partial \theta_{\mathcal{D}}}{\partial \theta_{\mathcal{F}}} - \lambda \cdot \frac{\partial L_{\mathcal{Q}}}{\partial \theta_{\mathcal{Q}}} \cdot \frac{\partial \theta_{\mathcal{Q}}}{\partial \theta_{\mathcal{F}}} \right), \tag{7}$$

where η denotes a learning rate. Implementation details are covered later in the Experiments section. The $\mathcal{Q}$ reduces the loss in the feed forward process in the direction of minimizing $L_{\mathcal{Q}}$. In backpropagation, the gradients of the quality discriminator, $\partial L_{\mathcal{Q}}/\partial\theta_{\mathcal{Q}}$, are updated in a standard backpropagation manner in $\mathcal{Q}$. On the other hand, it reversed negatively while they pass through the GRL. In other words, they are propagated $\mathcal{F}$ so that the feature extractor is trained in an adversarial way against $L_{\mathcal{Q}}$.

4. EXPERIMENTS

4.1 Datasets and Evaluation Metrics

For training, we employ DIV2K[23] train images. DIV2K is released for super-resolution (SR) and consists of 800 train images and 100 validation images. Since DIV2K is in a 2K resolution that maintains high quality even after compression, we reduced the resolution by a quarter to let image qualities be distinguishable when compressed. These resized original images are used as ground-truth. Then, we obtained the corresponding compressed dataset with JPEG image compression. Here, we create two compressed image sets of HQ and LQ using 70 and 10 QFs in the codec, respectively. We train the CARN with HQ and LQ as source and target, and vice versa. Since 70 and 10 QFs can lead to significantly different network parameters which are practically incompatible, this case (HQ $\rightarrow$ LQ) and its opposite case (LQ $\rightarrow$ HQ) are the most disadvantageous scenario for DA.

For evaluation, we use a DIV2K validation set, Set12,[3] and BSD68[24] to verify the applicability of the proposed method in different datasets. It is noteworthy that the data of Set12 and BSD68 are only used for test. All testing images are compressed in the same manner used for training set. We examine performance improvements in terms of three metrics: peak signal-to-noise ratio (PSNR), block-sensitive PSNR[25] (PSNR-B) and structural similarity[26] (SSIM).

4.2 Implementation Details

The proposed network is trained with randomly cropped 64×64 patches. Data augmentation is performed on training images, which includes random rotations of 90°, 180°, and 270° and flipping horizontally and vertically. The feature extractor and denoiser are pre-trained using the source quality. The parameters of the network are optimized using Adam[27] optimizer, and the learning rate is initially 10^{-4} and varies over time (500, 800 step).

4.3 Experimental Scenarios

We define various scenarios to verify the effectiveness of our proposed DA method. "same-quality (SQ)" refers to a scenario under the condition that a network is trained and tested using images of the same QF. This has a high possibility to be the best expectation of a network because a network can exploit the training and testing images of the same quality. In contrast, "different-quality (DQ)" refers to a scenario under that a network is trained with images of one quality and straightforwardly tested for images of another quality. Furthermore,"both-quality (BQ)" refers to a scenario under that a network is trained with images of both HQ and LQ. This is expected to have more robust denoising power as image samples with both qualities are exploited for training.

4.4 Results

We compared the experimental results for SQ, DQ, and BQ scenarios of the vanilla RDN in Tab. 1. The SQ scenarios provide improved results with vanilla RDN, and this phenomenon is observed both in QF=10 and 70. Meanwhile, the DQ scenarios have performance drop due to the discrepancies of training and testing QFs. The results showed 1.71 (dB) PSNR drop in the DQ scenario where the network is trained with $QF = 70$ and tested with $QF = 10$, compared to the SQ scenario for $QF = 10$. When the model is trained with $QF = 10$ and tested with $QF = 70$, we observe a substantial performance drop. For example, the PSNR, PSNR-B, and SSIM gains in a Set12 dataset are approximately -3.17 (dB), -2.59 (dB), and -0.16, indicating that there is severe performance degradation.

As clearly shown in Tab. 1, the proposed method achieves comparable or even better performance than the SQ scenarios which is expected to be the best performance. The best result appears in intra-dataset in which both the testing and training images are sampled from DIV2K. The results for PSNR, PSNR-B, and SSIM are 25.03 (dB), 25.01 (dB), and 0.47 for test $QF = 10$, and 29.26 (dB), 26.23 (dB), and 0.68 for test $QF = 70$, which

Table 1: Results in PSNR/PSNR-B/SSIM. In all training scenarios, denoiser ($\mathcal{D}$) takes 800 training images. In BQ scenario, 400 images from each QF are used while training (in total 800 images). The best result is in red color, and the second-best is in blue color. The JPEG rows indicate how compressed input images are. The result in parentheses is the difference for JPEG compressed images. A higher value indicates a higher gain. In our experiment, our train and test images are in grayscale and 8 bit. $\perp$ refers to the inter-dataset case, i.e., the configuration of the test dataset is different from that of the training dataset.

Method	Train Scenario	Quality Factor (QF)		Dataset								
				DIV2K			BSD68$^{\perp}$			Set12$^{\perp}$		
		Train	Test	PSNR	PSNR-B	SSIM	PSNR	PSNR-B	SSIM	PSNR	PSNR-B	SSIM
JPEG	-	-	10	23.99	23.77	0.42	26.96	26.50	0.42	27.65	27.20	0.38
Vanilla RDN	SQ	10		24.50(1.01)	24.98(1.21)	0.46(0.04)	28.14(1.18)	28.10(1.60)	0.46(0.04)	29.50(1.85)	29.48(2.28)	0.43(0.05)
	DQ	70		24.29(0.30)	24.16(0.39)	0.44(0.02)	27.39(0.43)	27.12(0.62)	0.44(0.02)	28.24(0.59)	28.00(0.81)	0.41(0.02)
	BQ	10, 70		24.95(0.96)	24.93(1.16)	0.46(0.04)	28.10(1.14)	28.05(1.55)	0.46(0.04)	29.44(1.78)	29.41(2.22)	0.43(0.04)
Ours	HQ→LQ	70		25.03(1.04)	25.01(1.24)	0.47(0.04)	28.14(1.18)	28.10(1.58)	0.46(0.04)	29.52(1.87)	29.50(2.30)	0.43(0.05)
JPEG	-	-	70	28.12	27.99	0.65	34.07	33.56	0.76	35.00	34.41	0.69
Vanilla RDN	SQ	70		29.15(1.03)	29.14(1.15)	0.68(0.03)	34.40(0.33)	34.35(0.80)	0.75(-0.01)	35.28(0.29)	35.25(0.83)	0.68(-0.01)
	DQ	10		27.19(-0.93)	27.19(-0.80)	0.58(-0.08)	31.11(-2.96)	31.11(-2.45)	0.62(-0.13)	31.83(-3.17)	31.82(-2.59)	0.53(-0.16)
	BQ	10, 70		29.09(0.97)	29.09(1.09)	0.68(0.02)	34.36(0.29)	34.31(0.75)	0.75(-0.01)	35.32(0.32)	35.28(0.87)	0.68(-0.01)
Ours	LQ→HQ	10		29.26(1.14)	29.23(1.24)	0.68(0.03)	34.39(0.33)	34.23(0.68)	0.76(-0.00)	35.27(0.27)	35.17(0.75)	0.68(-0.01)

are higher than SQ and BQ. We also observe remarkable improvements for inter-dataset in which testing samples are chosen from BSD68 and Set12, although the network has been trained using the DIV2K training dataset. In addition, our method shows better performance gain in SSIM, a metric considering human perception, than SQ and BQ. Although its gain is slightly below the best in PSRN and PSNR-B, it is noteworthy that the proposed method does not use the ground-truth during the adaptation. The results imply that the CARN trained with the proposed method becomes more robust to different levels of distortion in all the three datasets containing various characteristics of image samples.

In Tab. 2, we examine the performance of our method as the number of target samples changes and compare it to the BQ scenarios to analyze the effectiveness of our method with respect to the size and ratio of training data. "N_L" and "N_H" refer to the number of training images of low quality and high quality, respectively. We use $QF = 10$ as low quality and $QF = 70$ as high quality. In BQ scenarios, we fix the number of image samples of quality different from test quality to 400 and change the number of image samples with the same quality as test quality to 100, 400, and 800. For instance, as for test $QF = 10$, N_H is fixed to 400, and N_L is changed to 100, 400, and 800. Both vanilla RDN and our method show better performance as the number of samples with the same quality as the test samples increase. However, vanilla RDN requires additional training samples and corresponding ground-truths to learn $\mathcal{D}$. Meanwhile, in our method, it is notable that target samples are solely used for DA purpose in $\mathcal{Q}$ and are not used to train $\mathcal{D}$. Since target samples are not fed forward to $\mathcal{D}$, which accounts for most of the load in overall network training, our method takes less additional computation to vanilla RDN. The proposed method shows comparable or better performance than BQ of vanilla RDN over all range of ratios. In particular, our method prevails BQ when testing $QF = 70$ where input compressed image is already high quality, therefore, gain increment via compression artifact reduction is difficult. From the results, the highest gain in Our method outperforms the highest gain in BQ in respect to all metrics.

We provide qualitative results tested on $QF = 10$ in Fig. 3 and $QF = 70$ in Fig. 4. For each figure, (a)-(f) are the sample results of DIV2K, and (g)-(l) are the sample results of BSD68. Overall, the visual outcome mostly coincides with the numerical examination. In each figure, (a) and (g) are ground-truths of the sample images and (b) and (h) are the JPEG compressed sample images. The rest are reconstructed images obtained from the CARN in different training conditions. In Fig. 3 where testing $QF = 10$, we can see that our method Fig. 3c and 3i produce more visually pleasing results than all scenarios of the existing method Fig. 3d–3f and Fig. 3j–3l. As for DQ, Fig. 3e and 3k lag numerical results and produce erroneous visual results simultaneously. Meanwhile, the

Table 2: Experimental results while changing the ratio of the HQ and LQ images. N_L and N_H refer to the number of image samples with quality factor 10 and 70, respectively. * refers to the target data inputted to $\mathcal{F}$ and $\mathcal{Q}$ only and is not used to train Denoiser in a supervised manner. $\perp$ refers to the inter-dataset case, i.e., the configuration of the test dataset is different from that of the training dataset. The best result is red colored, and the second-best is blue colored. The JPEG rows indicate how compressed input images are. In our experiment, our train and test images are in grayscale and 8 bit.

Method	Train Scenario	Quality Factor (QF)				Dataset								
		# Train		Train	Test	DIV2K			BSD68$^{\perp}$			Set12$^{\perp}$		
		N_L	N_H			PSNR	PSNR-B	SSIM	PSNR	PSNR-B	SSIM	PSNR	PSNR-B	SSIM
JPEG	-	-	-	-	10	23.99	23.77	0.42	26.96	26.50	0.42	27.65	27.20	0.38
Vanilla RDN	BQ (ratio)	100	400	10 and 70	10	24.93(0.94)	24.91(1.14)	0.46(0.04)	28.08(1.11)	28.03(1.53)	0.46(0.04)	29.38(1.73)	29.36(2.16)	0.43(0.04)
		400				24.95(0.96)	24.93(1.16)	0.46(0.04)	28.11(1.14)	28.06(1.56)	0.46(0.04)	29.44(1.79)	29.41(2.22)	0.43(0.04)
		800				25.02(1.02)	24.99(1.22)	0.46(0.04)	28.15(1.19)	28.11(1.61)	0.46(0.04)	29.50(1.85)	29.48(2.28)	0.43(0.05)
Ours	HQ→LQ	100*	400	70		24.92(0.93)	24.90(1.13)	0.46(0.04)	28.05(1.08)	28.00(1.50)	0.46(0.04)	29.31(1.66)	29.29(2.09)	0.43(0.04)
		400*				24.97(0.97)	24.95(1.18)	0.46(0.04)	28.08(1.12)	28.04(1.54)	0.46(0.04)	29.45(1.80)	29.43(2.23)	0.43(0.04)
		800*				25.01(1.02)	24.99(1.22)	0.47(0.05)	28.14(1.18)	28.10(1.60)	0.47(0.05)	29.47(1.82)	29.45(2.25)	0.43(0.05)
JPEG	-	-	-	-	70	28.12	27.99	0.65	34.07	33.56	0.76	35.0	34.41	0.69
Vanilla RDN	BQ (ratio)	400	100	10 and 70		29.07(0.95)	29.07(1.07)	0.67(0.02)	34.34(0.27)	34.30(0.74)	0.75(-0.01)	35.29(0.29)	35.26(0.85)	0.68(-0.01)
			400			29.09(0.97)	29.09(1.09)	0.68(0.02)	34.36(0.29)	34.32(0.76)	0.75(-0.01)	35.35(0.35)	35.31(0.89)	0.68(-0.01)
			800			29.18(1.07)	29.17(1.18)	0.68(0.03)	34.38(0.31)	34.29(0.73)	0.75(-0.00)	35.30(0.30)	35.23(0.82)	0.68(-0.01)
Ours	LQ→HQ	400	100*	10		29.14(1.02)	29.13(1.14)	0.68(0.03)	34.43(0.36)	34.37(0.82)	0.76(-0.00)	35.29(0.29)	35.25(0.84)	0.68(-0.01)
			400*			29.16(1.04)	29.15(1.15)	0.68(0.03)	34.41(0.34)	34.34(0.79)	0.75(-0.01)	35.33(0.33)	35.27(0.86)	0.68(-0.01)
			800*			29.25(1.14)	29.24(1.24)	0.68(0.03)	34.48(0.41)	34.37(0.82)	0.76(-0.00)	35.44(0.45)	35.37(0.95)	0.68(-0.01)

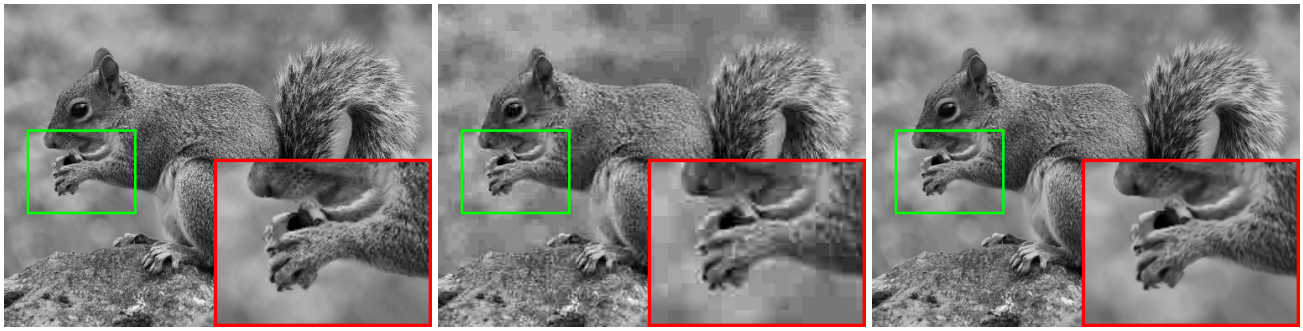
quantitative results for SQ and BQ are similar to ours. However, our method provides better perceptual visual quality than the vanilla RDN does under SQ in Fig. 3d and 3j and under BQ in Fig. 3f and 3l. For instance, in Fig. 3c, the texture of the squirrel fur is recovered in more detail. Moreover, the background is quite blurry in SQ and BQ, while our method preserves the background akin to the ground-truth. Fig. 4 can be interpreted in the same way. In Fig. 4, which is already visually gratifying since the testing $QF = 70$, quality improvements among SQ in Fig. 4d and 4j, BQ in Fig. 4f and 4l, and ours in Fig. 4c and 4i are indistinguishable. Nevertheless, given that the results in DQ are remarkably unsatisfactory, this proves that the existing CARN does not address different QFs.

5. CONCLUSION

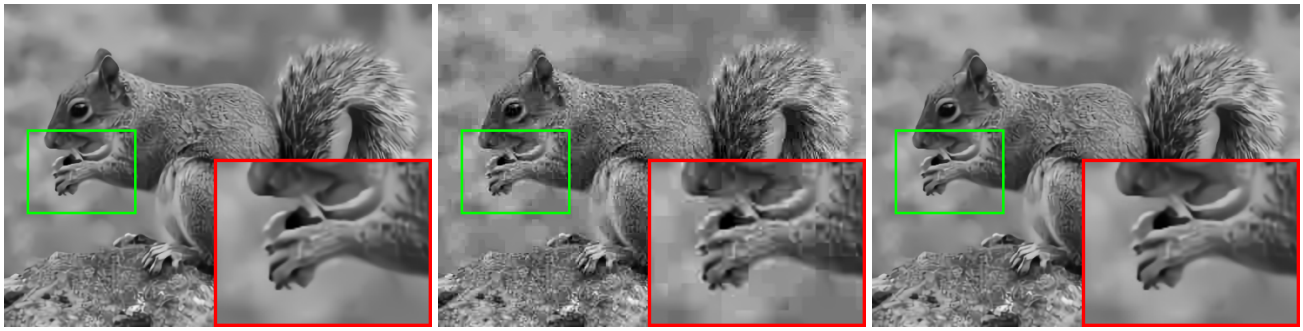
In conclusion, we introduced a novel training framework and perspective to consider images with different quality factors as different domains to develop a CARN to be more robust to various scenarios in which the model needed to encounter different quality factors in a target application. The domain adaptation was conducted by using adversarial learning to align different domains. Compared to existing studies, the proposed method can adapt the CARN to provide more robust results on various datasets and metrics. We also figure out that our DA method was robust to various configurations such as the size and ratio of training samples. Further studies need to be explored for training scheme to remove artifacts from compressed images with multi-quality or quality-agnostic images in a single network.

ACKNOWLEDGMENTS

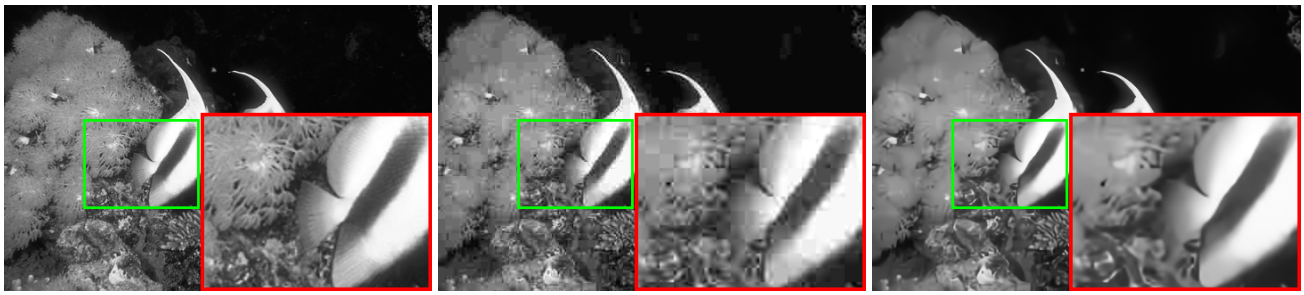
This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP–2021–2020–0–01460) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation)



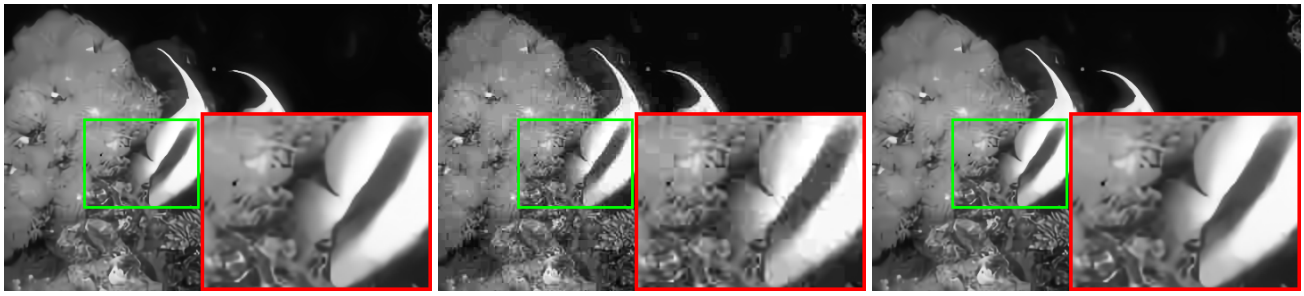
(a) **GT**: PSNR | PSNR-B | SSIM (b) **JPEG**: 23.72 | 23.51 | 0.37 (c) **Ours**: 24.13 | 24.10 | 0.43



(d) **SQ**: 24.11 | 24.08 | 0.43 (e) **DQ**: 23.92 | 23.80 | 0.39 (f) **BQ**: 24.12 | 24.10 | 0.43

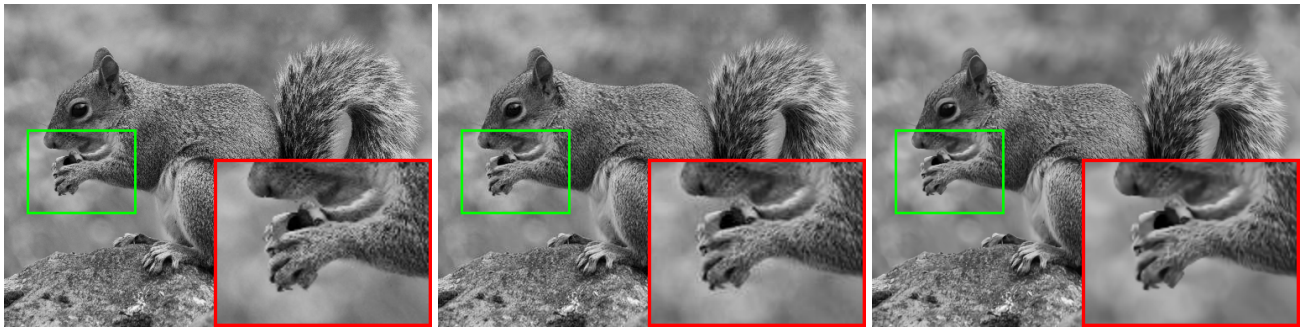


(g) **GT**: PSNR | PSNR-B | SSIM (h) **JPEG**: 27.39 | 27.00 | 0.40 (i) **Ours**: 28.44 | 28.38 | 0.42



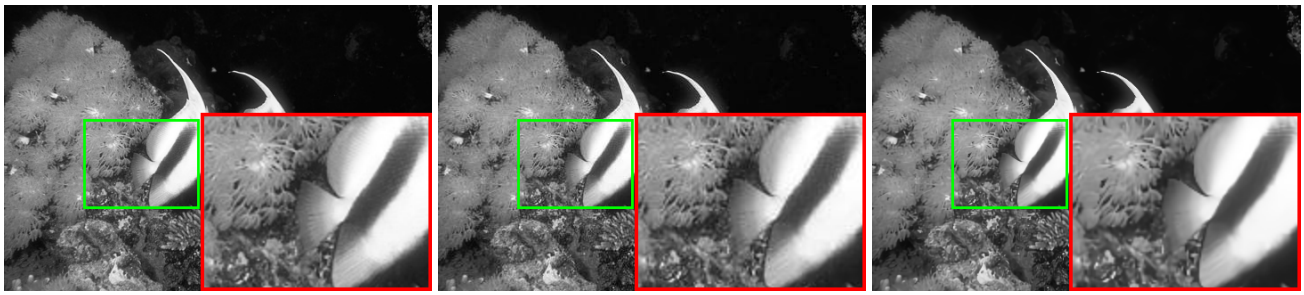
(j) **SQ**: 28.43 | 28.38 | 0.42 (k) **DQ**: 27.86 | 27.63 | 0.41 (l) **BQ**: 28.43 | 28.38 | 0.42

Figure 3: Sample images visualization corresponding to Tab. 1. (Tested on $QF = 10$) We provide a quantitative result in a format of "method: PSNR(dB) | PSNR-B(dB) | SSIM" for (b)-(f) and (h)-(l). The green box is an extension of the red box area for better visualizing results. Best viewed in a digital monitor.

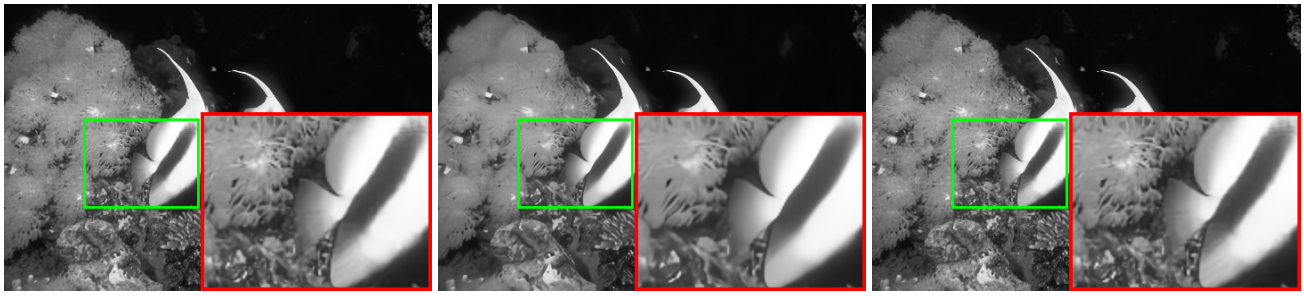


(a) **GT**: PSNR | PSNR-B | SSIM (b) **JPEG**: 27.31 | 27.25 | 0.69 (c) **Ours**: 27.63 | 27.62 | 0.70

(d) **SQ**: 27.58 | 27.58 | 0.70 (e) **DQ**: 26.05 | 26.05 | 0.60 (f) **BQ**: 27.59 | 27.59 | 0.70



(g) **GT**: PSNR | PSNR-B | SSIM (h) **JPEG**: 34.09 | 33.63 | 0.71 (i) **Ours**: 35.16 | 34.96 | 0.72



(j) **SQ**: 35.01 | 34.98 | 0.71 (k) **DQ**: 31.73 | 31.73 | 0.57 (l) **BQ**: 35.01 | 34.93 | 0.71

Figure 4: Sample images visualization corresponding to Tab. 1. (Tested on $QF = 70$) We provide a quantitative result in a format of "method: PSNR(dB) | PSNR-B(dB) | SSIM" for (b)-(f) and (h)-(l). The green box is an extension of the red box area for better visualizing results. Best viewed in a digital monitor.

REFERENCES

[1] Dong, C., Deng, Y., Loy, C. C., and Tang, X., "Compression artifacts reduction by a deep convolutional network," in [*Proceedings of the IEEE International Conference on Computer Vision (ICCV)*], (December 2015).

[2] Zhang, Y., Tian, Y., Kong, Y., Zhong, B., and Fu, Y., "Residual dense network for image restoration," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **43**(7), 2480–2495 (2021).

[3] Zhang, K., Zuo, W., Chen, Y., Meng, D., and Zhang, L., "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE transactions on image processing* **26**(7), 3142–3155 (2017).

[4] Tai, Y., Yang, J., Liu, X., and Xu, C., "Memnet: A persistent memory network for image restoration," in [*Proceedings of the IEEE international conference on computer vision*], 4539–4547 (2017).

[5] Dai, Y., Liu, D., and Wu, F., "A convolutional neural network approach for post-processing in hevc intra coding," in [*International Conference on Multimedia Modeling*], 28–39, Springer (2017).

[6] Lim, B., Son, S., Kim, H., Nah, S., and Mu Lee, K., "Enhanced deep residual networks for single image super-resolution," in [*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*], (July 2017).

[7] Borgwardt, K. M., Gretton, A., Rasch, M. J., Kriegel, H.-P., Schölkopf, B., and Smola, A. J., "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics* **22**(14), e49–e57 (2006).

[8] Ganin, Y. and Lempitsky, V., "Unsupervised domain adaptation by backpropagation," in [*International conference on machine learning*], 1180–1189, PMLR (2015).

[9] Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T., "Adversarial discriminative domain adaptation," in [*Proceedings of the IEEE conference on computer vision and pattern recognition*], 7167–7176 (2017).

[10] Meng, X., Chen, C., Zhu, S., and Zeng, B., "A new hevc in-loop filter based on multi-channel long-short-term dependency residual networks," in [*2018 Data Compression Conference*], 187–196, IEEE (2018).

[11] Jia, C., Wang, S., Zhang, X., Wang, S., Liu, J., Pu, S., and Ma, S., "Content-aware convolutional neural network for in-loop filtering in high efficiency video coding," *IEEE Transactions on Image Processing* **28**(7), 3343–3356 (2019).

[12] Zhang, Y., Shen, T., Ji, X., Zhang, Y., Xiong, R., and Dai, Q., "Residual highway convolutional neural networks for in-loop filtering in hevc," *IEEE Transactions on image processing* **27**(8), 3827–3841 (2018).

[13] Dong, C., Loy, C. C., He, K., and Tang, X., "Learning a deep convolutional network for image super-resolution," in [*European conference on computer vision*], 184–199, Springer (2014).

[14] Lin, W.-A., Liao, H., Peng, C., Sun, X., Zhang, J., Luo, J., Chellappa, R., and Zhou, S. K., "Dudonet: Dual domain network for ct metal artifact reduction," in [*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*], 10512–10521 (2019).

[15] Guo, J. and Chao, H., "Building dual-domain representations for compression artifacts reduction," in [*European Conference on Computer Vision*], 628–644, Springer (2016).

[16] Knaus, C. and Zwicker, M., "Dual-domain image denoising," in [*2013 IEEE International Conference on Image Processing*], 440–444, IEEE (2013).

[17] Kim, Y., Soh, J. W., and Cho, N. I., "Agarnet: adaptively gated jpeg compression artifacts removal network for a wide range quality factor," *IEEE Access* **8**, 20160–20170 (2020).

[18] Ehrlich, M., Davis, L., Lim, S.-N., and Shrivastava, A., "Quantization guided jpeg artifact correction," in [*Proceedings of the European Conference on Computer Vision*], Springer (2020).

[19] Hong, S., Im, W., Ryu, J., and Yang, H. S., "Sspp-dan: Deep domain adaptation network for face recognition with single sample per person," in [*2017 IEEE International Conference on Image Processing (ICIP)*], 825–829, IEEE (2017).

[20] Wang, H., Shen, T., Zhang, W., Duan, L.-Y., and Mei, T., "Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation," in [*European Conference on Computer Vision*], 642–659, Springer (2020).

[21] Ren, Z. and Lee, Y. J., "Cross-domain self-supervised multi-task feature learning using synthetic imagery," in [*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*], 762–771 (2018).

[22] Agresti, G., Schaefer, H., Sartor, P., and Zanuttigh, P., "Unsupervised domain adaptation for tof data denoising with adversarial learning," in [*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*], 5584–5593 (2019).

[23] Agustsson, E. and Timofte, R., "Ntire 2017 challenge on single image super-resolution: Dataset and study," in [*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*], 126–135 (2017).

[24] Martin, D., Fowlkes, C., Tal, D., and Malik, J., "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in [*Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*], **2**, 416–423, IEEE (2001).

[25] Yim, C. and Bovik, A. C., "Quality assessment of deblocked images," *IEEE Transactions on Image Processing* **20**(1), 88–98 (2010).

[26] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P., "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing* **13**(4), 600–612 (2004).

[27] Kingma, D. P. and Ba, J., "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980* (2014).