One-Two-One Networks for Compression Artifacts Reduction in Remote Sensing

Baochang Zhang[a], Jiaxin Gu[a], Chen Chen[c], Jungong Han[d], Xiangbo Su[a], Xianbin Cao[b], Jianzhuang Liu[e]

[a]*School of Automation Science and Electrical Engineering*
Beihang University, Beijing, China
[b]*School of Electronics and Information Engineering*
Beihang University, Beijing, China
[c]*Center for Research in Computer Vision (CRCV), University of Central Florida, Orlando, FL, USA*
[d]*School of Computing and Communications, Lancaster University, LA1 4YW, UK.*
[e]*Noah's Ark Lab, Huawei Technologies Co. Ltd., China*

Abstract

Compression artifacts reduction (CAR) is a challenging problem in the field of remote sensing. Most recent deep learning (DL) based methods have demonstrated superior performance over the previous hand-crafted methods. In this paper, we propose an end-to-end one-two-one (OTO) network, to combine different deep models, i.e., summation and difference models, to solve the CAR problem. Particularly, the difference model motivated by Laplacian pyramid is designed to obtain the high frequency information, while the summation model aggregates the low frequency information. We provide an in-depth investigation into our OTO architecture based on the Taylor expansion, which shows that these two kinds of information can be fused to gain more capacity of handling complicated image compression artifacts, especially the blocking effect in compression. Extensive experiments are conducted to demonstrate the superior performance of the OTO networks, as compared to the state-of-the-arts on remote sensing datasets and other benchmark datasets. The source code will be available here[1].

Keywords: Compression Artifacts Reduction, Remote Sensing, Deep Learning, One-Two-One Network

2010 MSC: 00-01, 99-00

---

[1]https://github.com/bczhangbczhang/

1. Introduction

In remote sensing, the satellite- or aircraft-based sensor technologies are used to capture and detect objects on Earth. Thanks to various propagated signals (e.g., electromagnetic radiation), remote sensing makes the data collection from dangerous or inaccessible areas possible, and therefore plays a significant role in many applications including monitoring, military information collection and land-use classification [1, 2, 3, 4]. With the technological development of various satellite sensors, the volume of high-resolution remote sensing image data is increasing rapidly. Hence, proper compression of the satellite image becomes essential, which enables information exchange much more efficient, given a limited band width.

Existing compression methods generally fall into two categories: lossless (e.g., PNG) and lossy (e.g., JPEG) [5]. The lossless methods usually provide better visual experience to users, but lossy methods often achieve higher compression ratios via non-invertible compression functions along with trade-off parameters to balance the data amount and the decompressed quality. Therefore the lossy compression schemes are always preferred by consumer devices in practice due to higher compression rate [5]. However, high compression rate comes with the cost of having compression artifacts on the decoded image, which is a barrier for many applications, such as image analysis. Therefore, there is a clear need for compression artifact reduction, which is able to gain visual quality of the decompressed image, which can influence the visual effect and low-level vision processing [6].

In theory, the compression artifacts are in relation to the schemes used for compression. Take JPEG compression as an example, blocking artifacts are caused by discontinuities at the borders when encoding adjacent 8×8 pixel blocks, which are in the form of ringing effects and blurring due to the coarse quantization of the high frequency components. To deal with these compression artifacts, an improved version of JPEG, named JPEG 2000, is proposed, which adopts the wavelet transform to avoid blocking artifacts, but still undergoes ringing effects and blurring. As an excellent alternative, SPIHT [7] showed that using simple uniform scalar quantization, rather than complicated vector quantization, also yields superior results. Due to its simplicity, SPIHT has been successful on natural (portraits, landscape, weddings, etc.) and medical (X-ray, CT, etc.) images. Furthermore, its embedded encoding process has proved to be effective in a broad range of reconstruction qualities. For instance, it can code fair-quality portraits and high-quality medical images equally well (as compared with other methods in the same conditions). However, in the field of remote sensing, the images usually suffer from severe artifacts after compression as shown
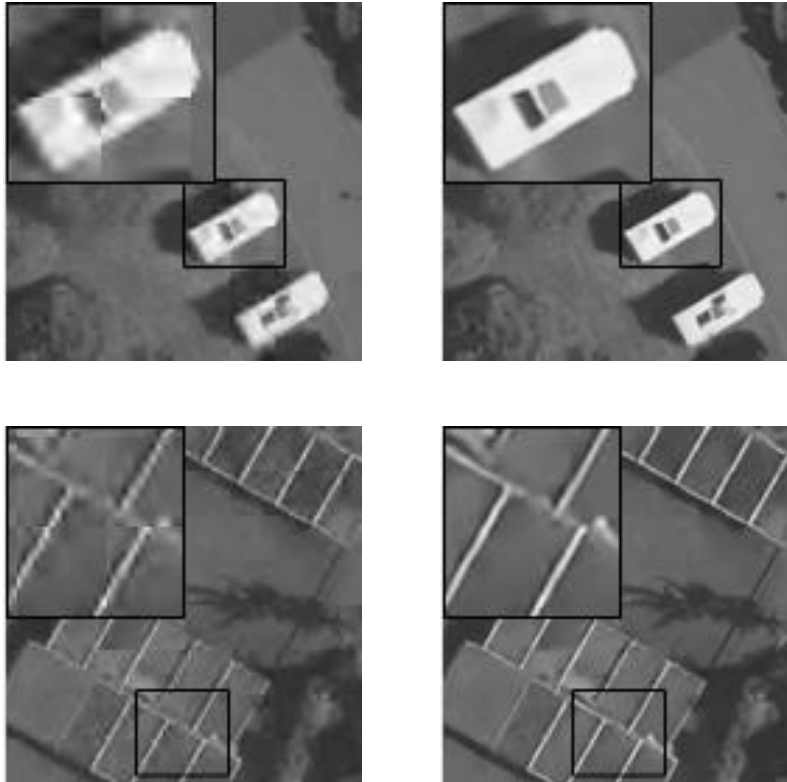
2

Figure 1: Left: the SPIHT-compressed remotely sensed images with obvious blocking artifacts. Right: the restored images by our OTO network, where lines are sharp and blurring is removed.

in Fig. 1, which poses challenges to many high-level vision tasks, such as object detection [8, 9], classification [1, 10], and anomaly detection [11].

To cope with various compression artifacts, many conventional approaches have been proposed, such as filtering approaches [12], [13], [14], specific priors (e.g., the quantization table in DSC [15]), and thresholding techniques [16, 17]. Inspired by the great success of deep learning technology in many image processing applications, researchers start to exploit this powerful tool to reduce the compression artifact. Specifically, the Super-Resolution Convolutional Neural Network (SR-CNN) [18] exhibits great potential of an end-to-end learning in image super-resolution. It is also pointed out that conventional sparse-coding-based image restoration model can be equally seen as a deep model. However, if we directly apply SRCNN to the compression artifact reduction task, the features extracted by its first layer are noisy, which will cause undesirable noisy patterns in

reconstruction. Thus the three-layer SRCNN is not suitable for compressed image restoration, especially when dealing with complex artifacts. Thanks to transfer learning, ARCNN [6] has been

⁴⁵ successfully applied to image restoration tasks. However, without exploiting the multi-scale information, ARCNNs fail to solve more complicated compression artifact problems. Although many deep models with different architectures have been explored (e.g., [18, 6, 19]) to solve the artifact reduction problem, there is little work incorporating different models in a unified framework to inherit their respective advantages.

⁵⁰   In this paper, a generic fusion network, dubbed as one-two-one (OTO) network, is developed for complex compression artifacts reduction. The general framework of the proposed OTO network is presented in Fig. 2. Specifically, it consists of three sub-networks: a normal-scaled network, a small-scaled network with max pooling to increase the network receptive field, and a fusion network to perform principled fusion of the outputs from the summation and difference models. The summation

⁵⁵ model aggregates the low frequency information captured from different network scales, while the difference model is motivated by the Laplacian pyramid which is able to describe the high frequency information (such as edges). By combining the summation and difference models, both low and high frequency information of the image can be better characterized. This is motivated by the fact that adopting different schemes to process high frequency and low frequency information always benefits

⁶⁰ to low-level image processing applications, such as image denoising [20] and image reconstruction [21]. Most importantly, we provide an in-depth investigation into our OTO architecture based on the Taylor expansion, which shows that these two kinds of information are fused in a nonlinear scheme to gain more capacity to handle complicated image compression artifacts. From a theoretical perspective, this paper proposes a principled combination of different CNN models, providing the

⁶⁵ capability of coping with the extremely challenging task of the large blocking effect. Extensive experimental results verify that combining diverse models effectively boosts the performance. In a summary, we have the following contributions in this paper.

1. We develop a new one-two-one (OTO) network, to combine different models based on an end-to-end deep framework aiming to effectively deal with complicated artifacts, i.e., the big
⁷⁰   blocking effect in compression.

2. We are motivated by the idea of Laplacian pyramid, which is extended in the deep learning framework, and explicitly used to capture the high frequency information in images. We show in the experiments that the difference model is able to effectively improve the compression

4

Table 1: A brief description of variables used in the paper.

Y: input compressed image	F_1: normal-scaled network
$\tilde{Y}$: output of the first convolutional layer	F_2: small-scaled network
N_1: output of F_1	N_2: output of F_2
Sum: summation of N_1 and N_2	Dif: difference between N_1 and N_2
G_S: summation model	G_D: difference model
H_S: output of G_S	H_D: output of G_D
F_3: fusion network	α: weight term between the two sub-networks
$G'_S(\cdot)$: derivative of G_S	$G'_D(\cdot)$: derivative of G_D
γ: constant term	$o[\cdot, \cdot]$: higher order infinitesimal
X: uncompressed target image	

artifact reduction performance.

3. Based on the Taylor expansion, we lead to two OTO variants, which provide a profound investigation into our method.

4. Extensive experiments are conducted to validate the performance of OTO over the state-of-the-arts on both the benchmark datasets and remote sensing datasets.

For ease of explanation, we summarize main variables in Table 1. The rest of the paper is organized as follows. Section 2 introduces the related works, and Section 3 describes the details of the proposed method. Experiments and results are presented in Section 4. Finally, Section 5 concludes the paper.

2. Related work

The OTO network is proposed to combine summation and difference models in the end-to-end framework. Particularly, the difference model motivated by the Laplacian pyramid is designed to obtain high frequency information, while the summation model aggregates the low frequency information. Compared to the summation model, the difference model can provide more details information. In this section, we briefly described the related work about how the high frequency information used in low-level image processing, and also the previous CAR methods.

On the high frequency information. High frequency information has been exploited in tasks such as pansharping [22], superresolution [23] and denoising [24]. However, the way of exploring it is different from ours. More specifically, in image superresolution, a low resolution input image is first interpolated to have the same size of the high resolution image as input. Then the goal of the

network becomes learning the high resolution image from the interpolated low resolution image [23]. In other words, the network essentially aims to learn the high frequency information in order to obtain the high resolution output [25, 26]. In pansharpening, high frequency details are not available for multispectral bands, and must be inferred through the model [27, 28, 22] starting from those of Pan images. In denoising, residual learning is utilized to speed up the training process as well as boost the denoising performance [29, 20, 24]. The Laplacian pyramid is ubiquitous for decomposing images into multiple scales and is widely used for image analysis [30, 31], which is computed as the difference between the original image and the low pass filtered image. This process is continued to obtain a set of band-pass filtered images, since each is the difference between two levels of the Gaussian pyramid. Laplacian pyramids have been used to analyze images at multiple scales for a broad range of applications such as compression [31], texture synthesis [32], and harmonization [33].

Traditional CAR methods. Traditional methods for compression artifacts reduction are generally categorized into deblocking-based and dictionary-based algorithms. The deblocking-based algorithms mainly focus on removing blocking and ringing artifacts using filters in the spatial domain or utilizing wavelet transforms and setting thresholds at different wavelet scales in the frequency domain. Among them, the most successful work is Shape-Adaptive Discrete Cosine Transformation (SA-DCT) [17], which achieved the state-of-the-art performance during the 2000s. However, similar to other deblocking-based methods, SA-DCT suffers from blurry edges and smooth texture regions as well. It is worth noting that SA-DCT is an unsupervised method, which is more powerful than supervised methods when there are not enough samples available. The supervised dictionary-based algorithms, such as RTF [34], S-D2 [15], take compression artifacts reduction as a restoration problem and reverse the impact of DCT-domain quantization by learned dictionaries. Unfortunately, the optimization procedure of sparse-coding-based approaches is always complicated and the end-to-end training does not seem to be possible, which limits their reconstruction performance.

Deep CAR methods. Recently, deep convolutional neural networks have shown promising performance on both high-level vision tasks, such as classification [35, 36], detection [37, 38, 39, 40] and segmentation [41, 42, 43], and low-level image processing like super-resolution. Super-Resolution Convolutional Neural Networks (SRCNN) [18] utilize a three-layer CNN to increase the resolution of images and achieve superior results over the traditional SR algorithms like A+ [44]. Following the idea of SRCNN, Yu et al. [6] eliminate the undesired noisy patterns by directly applying SRCNN architecture for compression artifacts suppression and prove that transfer learning

6

also succeeds in low-level vision problems. Compression artifacts reduction CNN [6] mainly benefits from transfer learning in three aspects: from shallow networks to deep networks, from high-quality training datasets to low-quality ones and from one compression scheme to another scheme. Svoboda et al. [45] learn a feed-forward CNN by combining residual learning, skip architecture and symmetric weight initialization to improve image restoration performance. The generative adversarial network (GAN) is also successfully used to solve the CAR problem. In [46], the Structural Similarity (SSIM) loss is devised, which is a better loss with respect to the simpler Mean Squared Error (MSE), to re-formulate the compression artifact removal problem in a generative adversarial framework. The method obtains better performance than MSE trained networks.

Due to the fixed quantization table in the JPEG compression standard, it is reasonable to take advantage of JPEG-related prior for better restoration performance. Deep Dual-domain Convolutional neural Network (DDCN) [47] adds DCT-domain prior into the dual networks so that the network is able to learn the difference between the original images and compressed images in both pixel-domain and DCT-domain. Likewise, D3 method [48] converts sparse-coding approaches into an LISTA-based [49] deep neural network, and gains both speed and performance. Both of DDCN and D3 adopt JPEG-related priors to improve reconstruction quality. One-to-many network [50] is proposed for compression artifacts reduction. The network consists of three losses, a perceptual loss, a naturalness loss, and a JPEG loss, to measure the output quality. By combining multiple different losses, the one-to-many network is able to achieve visually pleasing artifacts reduction.

Challenges of the CAR problem. In spite of already achieving good compression artifact removal performance, they still have limitations, especially when dealing with satellite imagery. Prior-based methods may not be generalized to other compression schemes like SPIHT, and therefore their applications are limited for the reason that satellite- or aircraft-based sensor technologies use variable compression standards. Another ignored problem-specific prior is the size of blocks, which is typically 8×8. The existing JPEG-based methods crop images into sub-samples or patches with small size like 32×32 and use 8×8 blocks for processing. However, larger block size like 32×32 is often adopted in the digital signal processor (DSP) of satellites for parallel processing. In this case, an image patch only contains a whole block and might have negative impact on the training process. As a result, it is important for sub-samples to contain several blocks so that the networks can perceive the spatial context between adjacent blocks. On the other hand, the existing deep learning based compression artifact removal approaches mainly focus on the architecture de-

7

sign [6, 18, 45] or changing the loss function [46, 50], with no theoretical explanations so that they fail to provide more profound investigation into methodologies. Moreover, the benefits of different network architectures are not fully explored for solving the CAR problem.
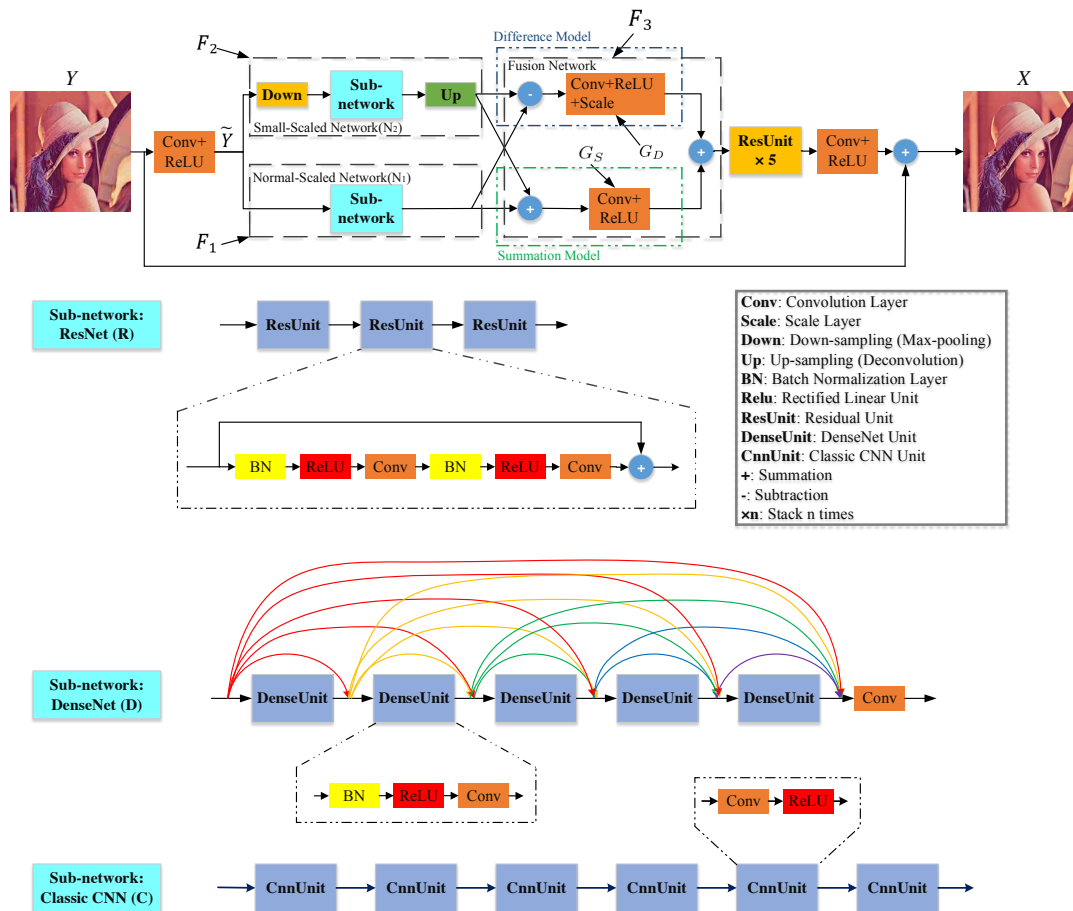

Figure 2: The architecture of One-Two-One (OTO) Network. Two different CNN models are combined in a principled framework, the outputs of which are further processed based on a fusion network. The details of the three sub-networks are also included.

3. One-Two-One Networks

The OTO networks are designed to reduce compression artifacts based on a unified framework. As shown in Fig. 2, two different models (summation model and difference model) are used to

8

restore the input image individually, whose advantages are inherited by a CNN fusion network, and thus leading to a better performance than using each of them individually. In what follows, we address two issues to build the OTO network. We first describe the motivation of OTO, along with a theoretical investigation into the network architecture which leads to two variants. We then elaborate the structures of the proposed OTO network, which are divided into three specific sub-networks. For each of them, we give the details of the implementation.

3.1. Theoretical Investigation of OTO

OTO is a general framework aiming to combine different deep models of different architectures. In OTO, a hierarchical CNN structure is exploited to capture multi-scaled texture information, which is very effective in dealing with various compression artifacts. In addition, each network in our framework carries out a specific objective, i.e., different-scaled textures, and we end up combining them together to obtain better results. The idea origins from the Laplacian pyramid for capturing edge information, but we use the different scaled networks to implement the idea in the deep learning framework. The small-scaled network involves spatial max pooling, which essentially increases the network receptive field and aggregates information in larger spatial area. Therefore, by combining small-scaled network and normal-scaled network features, the network learns features from different scales. Inspired by the Laplacian pyramid, the difference model is exploited in the deep framework and able to describe high frequency information, while the summation model captures low frequency information. We then combine both in a principled end-to-end deep framework. We like to highlight our idea from a more basic way. We provide a sensible way to combine the low and high frequency information in the deep learning framework, and also theoretically explain it with the Taylor expansion. In OTO, we have:

$$N_1 = F_1\left(\tilde{Y}\right), \tag{1}$$

and

$$N_2 = F_2\left(\tilde{Y}\right), \tag{2}$$

where $\tilde{Y}$ is the output of the first convolutional network, which is designed to pre-process the input compressed image Y based on convolution layers. N_1 and N_2 denote the outputs of the two branch networks, i.e., normal-scaled network and small-scaled network. To better restore the input image

9

X, we exploit two different networks, i.e., summation model and difference model, based on $\tilde{Y}$, which complement each other in terms of different network architectures. The summation model is used to mitigate the disparity between two networks, while the difference model highlights that different CNNs are designed for different purposes in order to obtain better restoration results. We have:

$$Sum = N_1 + N_2, \tag{3}$$

which actually aggregates the low frequency information.

$$Dif = N_1 - N_2, \tag{4}$$

which describes the high frequency information as shown in the Laplacian pyramid. G_S and G_D denote the two branches following the summation and subtraction operation in Fig. 2 respectively. Both kinds of information are then combined together for a better restoration of the input, and we have:

$$H_S = G_S\left(Sum\right), \tag{5}$$

and

$$H_D = G_D\left(Dif\right), \tag{6}$$

where H_S and H_D are the outputs of the two branches. They are then combined together via a nonlinear operation, which is designed to be robust to the artifacts in the compressed images. And we have:

$$
\begin{aligned}
F_3\left(\tilde{Y}\right) &= H_S + \alpha H_D = G_S\left(Sum\right) + \alpha G_D\left(Dif\right) \\
&= G_S\left(N_1 + N_2\right) + \alpha G_D\left(N_1 - N_2\right)
\end{aligned}
\tag{7}
$$

10

where α is a weight factor to balance different models. Based on Taylor expansion on G_S and G_D, we prove that our OTO is actually the combination of N_1 and N_2 based on a nonlinear scheme as:

$$
\begin{aligned}
F_3\left(\tilde{Y}\right) =& G'_S\left((N_1 + N_2)^*\right) \times (N_1 + N_2) + \alpha G'_D\left((N_1 - N_2)^*\right) \times (N_1 - N_2) \\
& + \gamma + o\left[(N_1 + N_2), (N_1 - N_2)\right] \\
=& \left(G'_S\left((N_1 + N_2)^*\right) + \alpha G'_D\left((N_1 - N_2)^*\right)\right) N_1 + \\
& \left(G'_S\left((N_1 + N_2)^*\right) - \alpha G'_D\left((N_1 - N_2)^*\right)\right) N_2 + \gamma + o\left[(N_1 + N_2), (N_1 - N_2)\right],
\end{aligned}
\tag{8}
$$

where $*$ means that there is a point, which is always differential, used in Taylor expansion. γ denotes the constant term, and $o[(N_1 + N_2), (N_1 - N_2)]$ denotes the higher order infinitesimal. More specifically, $o[(N_1 + N_2), (N_1 - N_2)]$ in Eq. 8 is the non-linear part and the remaining parts are linear parts. Note that the adopted nonlinear OTO model includes both the linear and nonlinear parts.

Based on Eq. 8, two linear OTO variants can be obtained as shown in Fig. 3 and Fig. 4. The first one, termed as OTO(Linear), is:

$$
F_{3,1}\left(\tilde{Y}\right) = N_1 + \alpha N_2,
\tag{9}
$$

which can be derived from the linear part of Eq. 8. In its implementation we learn α that is elaborated in the experimental part. Particularly $\alpha = 1$, we obtain the second one:

$$
F_{3,2}\left(\tilde{Y}\right) = N_1 + N_2,
\tag{10}
$$

which leads to our baseline, termed as OTO(Sum).

3.2. The architectures of OTOs

There are three distinct parts in OTOs: a) normal-scaled restoration network, b) small-scaled restoration network, c) fusion network. For sub-networks a) and b), three kinds of CNN models are available: R, D and C (short for ResNet, DenseNet and Classic CNNs respectively). The details about the OTO network are shown in Fig. 2.

ResNet(R): For each ResUnit, we follow the latest variant proposed in [51], which is more powerful than its predecessors. More specifically, in each ResUnit, batch normalization layer [52],
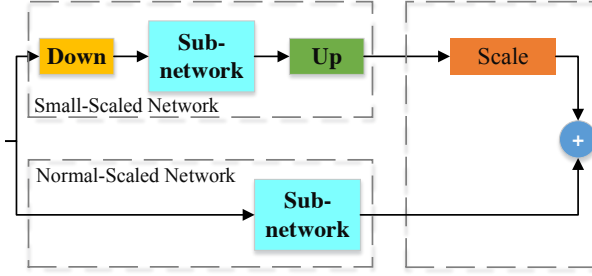
11

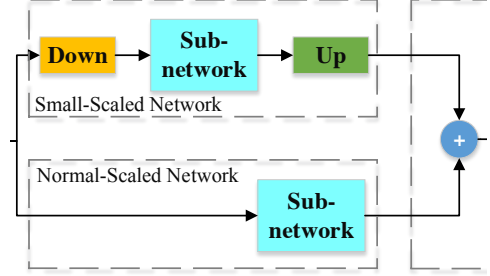Figure 3: The architecture of OTO(Linear) with a learned α to balance the two branch networks.

Figure 4: The architecture of OTO(Sum) without the difference model.

ReLU layer [53] and convolution layer are stacked twice in sequence.

DenseNet(D): Inspired by Densely Connected Convolutional Networks [54], to further improve the information flow between layers we propose a different connectivity pattern: we introduce direct connections from any layer to all subsequent layers. In DenseNet, the feature fusion method is converted from addition to concatenation compared with ResNet, resulting in wider feature maps. The growth rate k is an important concept in DenseNet which means how fast the width of feature maps grows and in our implementation, we set k to 8. For each DenseUnit, we also follow the pre-activation style unit as ResUnit except the number of convolutional layers is reduced to 1. As can be seen in Fig. 2, five DenseUnits are stacked sequentially followed by a convolutional layer to reduce the width of feature map so that it can be fused with the other sub-network.

Classic CNNs(C): The classic CNN models only take advantages of convolutional layers and activation layers. The CnnUnit consists of one convolutional layer and one ReLU layer, and 6 CnnUnits are stacked to form the Classic CNN sub-network.

In the sub-network b), we utilize 2×2 max-pooling to decrease the size of feature map by half, which obtains the following benefits: the computational cost is decreased to $1/4$, and with more robust features extracted compared to the sub-network a), and thus enlarging the perceptional field.

Fusion Network: Following Eq.7, we construct the fusion network. Convolutional layers with ReLUs serve as the non-linear operation, and scale layers serve as the auto-learned term. After fusion, we stack 5 more ResUnits to further restore the images.

OTO Naming Rules: For convenience, we use abbreviations to represent the three kinds of sub-networks. The first and second abbreviations after OTO represent the normal-scaled and

small-scaled sub-networks respectively. For example, OTO_RD stands for an OTO network whose normal-scaled sub-network is a ResNet and whose small-scaled sub-network is a DenseNet.

Multi-scale OTO Networks: To further investigate our proposed OTO network, we design a multi-scale network whose structure is shown in Fig. 5. $\times \frac{1}{2}$ and $\times \frac{1}{4}$-scale features are first fused by the first fusion network to get the combined $\times \frac{1}{2}$-scale feature. Then the fused feature along with the $\times 1$-scale feature serves as the input of the second fusion network. Except for the architecture, all the other details are the same as the two-scale OTO network.

It should be noted that the proposed OTO network exploits a series of ResUnits to fit the residual of the input and target images. In other words, there is a long and direct shortcut connecting the input image and the output of the subsequent network apart from the identity shortcut of each ResUnit. VDSR [23] has already proved that learning the residual between low-resolution and high-resolution image is more efficient and effective in the super-resolution task, because the difference is small, i.e., residual is sparse. In CAR task, this intuition is tenable because the compression algorithms do not change the essence of the image. As a result, the ResUnit leads to a sparse residual and thus we can train the network efficiently.
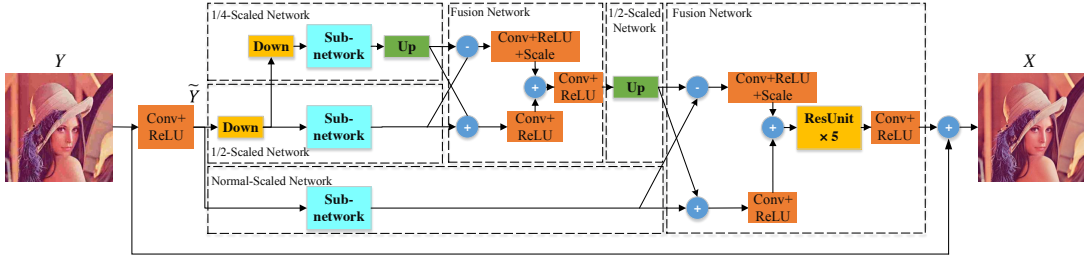
Figure 5: The architecture of multi-scaled OTO network (OTO_RRR) in which $\times 1, \times \frac{1}{2}$ and $\times \frac{1}{4}$—scaled features are exploited. The sub-networks are the same as in Fig. 2.

4. Implementation and Experiments

4.1. Datasets

In order to evaluate the OTO network, three groups of training and test dataset settings are designed, which are given according to their different test sets.

LIVE1 and Classic 5: Following the protocol of ARCNN [6], we evaluate our proposed network based on BSD500 [55], where the training and test set are combined to form a 400-image training

set and a 100-image validation set. The disjoint dataset LIVE1 [56] containing 29 images is chosen as our test set. Another test set we adopt is Classic 5, one of the most frequently used datasets for evaluating the quality of images.

230 **BSD500 Test set:** The BSD500 test set contains 200 images, which is more challenging than LIVE1, and more widely adopted in the recent research papers [47]. Considering that the 200-image BSD500 training set is too small to generate enough sub-samples when a large stride is chosen, we perform data augmentation by rotating the original images by 90, 180 and 270 degrees. The remaining 100-image BSD500 validation set is used for validation.

235 **Remotely Sensed Datasets:** There are two public remote sensing datasets on "ISPRS Test Project on Urban Classification and 3D Building Reconstruction": "Downtown Toronto" and "Vaihingen" [57]. To validate the performance of OTO on remote sensing images, "Downtown Toronto" dataset is employed, which contains various landscapes, such as ocean, road, house, vehicle and plant. To build a dataset for the compression artifact reduction problem, we preprocess the high-
240 resolution images in the "Downtown Toronto" dataset by using various compression algorithms, but obviously without the need for labeling the ground truth. SPIHT compression algorithm is used. The SPIHT algorithm can be applied to satellite images, where the original images is cropped into sub-images with a specific size 32×32. Compared to JPEG, the size of block artifacts in SPIHT is 32×32, which is much larger than that used in JPEG. It is different from the quality factor in JPEG
245 that the compression degree is decided by compression ratio, such as 8, 16, 32 and 64. Afterwards, we build the datasets used for training and validation. We randomly pick up 400 non-overlapping sub-images from the source images and the compressed images to form the training set and each image has a uniform size of 512×512. Then we do the same operation to get the 200-image disjoint validation set. For testing, we use the other dataset "Vaihingen" to build a 400-image test set that
250 has the same setting as the training set.

Evaluation Metrics: To quantitatively evaluate the proposed method, three widely used metrics: peak signal-to-noise ratio (PSNR), PSNR-B [58] and structural similarity (SSIM) [59] are adopted in our experiments. PSNR is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its rep-
255 resentation, which is most commonly used to measure the quality of reconstructed image after a lossy compression. The PSNR-B modifies PSNR by including a blocking effect factor resulting in a better metrics than PSNR for quality assessment of impaired images. SSIM index is a method for

14

predicting the perceived quality of digital images. SSIM considers image degradation as perceived change in structural information. While incorporating important perceptual phenomena, it also includes both luminance masking and contrast masking terms.

Other Settings: We only focus on restoring the luminance channel of the compressed image, and RGB-to-YCbCr operation is applied via MATLAB function. We also use MATLAB to carry out JPEG compression to generate compressed images with different qualities, such as QF-10, 20, 30 and 40. It is also worth noting that we crop every image such that the number of pixels in height and width are even since an odd number will affect the process of down-sampling and up-sampling (padding is necessary). To train the proposed OTO network, we choose SGD as the optimization algorithm with a momentum 0.9 and a weight decay 0.001. The initial learning rate is 0.01 with a degradation of 10% over every 30000 iterations before it reaches the maximum iteration number 120000.

4.2. Sub-networks and Multi-scaled network

Table 2: Results on different combination of sub-networks. Red marks mean the best results and blue marks mean the second best results

LIVE1	Quality	OTO_CC	OTO_CR	OTO_RC	OTO_DD	OTO_DR	OTO_RD	OTO_RR	OTO_RRR
PSNR	10	29.01	29.25	29.24	29.25	29.24	29.23	29.28	**29.38**
	20	31.39	31.61	31.61	31.63	31.63	31.61	31.67	**31.79**
	30	32.52	33.04	33.05	33.04	33.06	33.07	33.08	**33.10**
	40	33.56	34.07	34.08	34.09	34.09	34.09	34.10	**34.13**
SSIM	10	0.8213	0.8289	0.8293	0.8293	0.8296	0.8293	0.8298	**0.8301**
	20	0.8851	0.8946	0.8949	0.8952	0.8951	0.8950	0.8954	**0.8958**
	30	0.9124	0.9212	0.9217	0.9215	0.9218	0.9218	**0.9218**	**0.9218**
	40	0.9274	0.9358	0.9361	0.9361	0.9362	0.9362	0.9362	**0.9365**
PSNR-B	10	28.74	28.94	28.92	28.95	28.92	28.91	28.95	**29.13**
	20	31.03	31.09	31.14	31.15	31.14	31.14	31.17	**31.29**
	30	31.67	32.46	32.47	32.44	32.45	32.45	32.48	**32.52**
	40	33.04	33.40	33.43	33.44	33.44	33.45	33.48	**33.53**

As mentioned before, the OTO network is a framework that can take advantage of any CNNs, e.g., ResNet(R), DenseNet(D) and Classic CNN(C), as its sub-networks. The results of combining different kinds of sub-networks are shown in Table 2. Classic CNNs obtain the worst results, but which can be improved by using different scales (OTO_CC) or combining with ResNet (OTO_CR and OTO_RC). The OTO based on the densely connected network (OTO_DD) is designed to encourage feature reuse, but the lack of an identity mapping enforces the network to learn residual, which deems its failure. The combination of DenseNet and ResNet with different scales (OTO_DR

and OTO_RD) are affected by two kinds of discriminated features. In contrast, residual learning benefits more on the CAR problem, and the combination of two ResNets (OTO_RR) outperforms all other combinations. Multi-scaled features show promising results, and we design a multi-scaled OTO network (OTO_RRR) by adding an 1/4-scaled sub-network to OTO_RR. The result outperforms OTO_RR with a large margin on all three metrics. Even though OTO_RRR has outstanding performance, its computational cost increases almost 25%, resulting in more training and test time. After evaluating the pros and cons, we choose OTO_RR as our main framework and if not mentioned, OTO means OTO_RR in the following. We further design an experiment by removing one of the sub-networks each time to investigate the function of the sub-networks. The results in Table 3 indicate that the normal-scaled feature is shown to be more helpful than the small-scaled feature when only one sub-network is adopted.

Table 3: Experiments on the sub-networks.

LIVE1	Algorithm	PSNR	PSNR-B	SSIM
	OTO(normal-scaled)	28.65	27.91	0.8263
QF-10	**OTO(small-scaled)**	28.26	27.62	0.8245
	OTO	**29.28**	**28.95**	**0.8298**

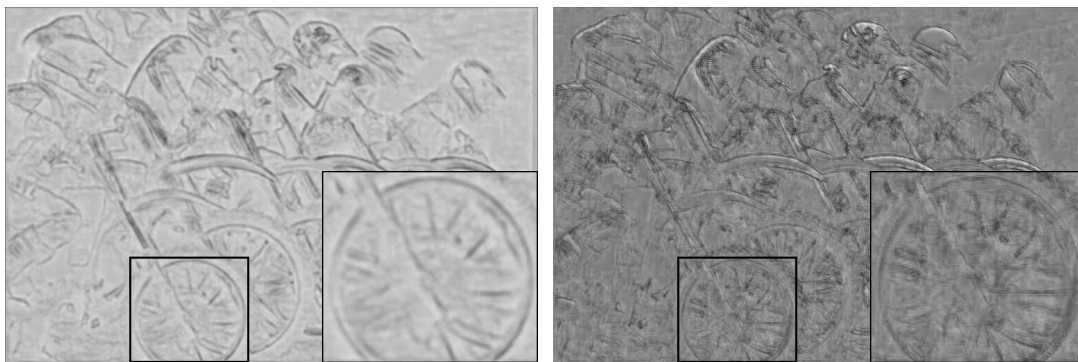Figure 6: Left: the feature map of the summation model, which describes the low frequency information. Right: the feature map of the difference model, which describes the high frequency information.

4.3. OTO vs. its two Variants

As mentioned above, OTO has the capability to utilize the nonlinear model, i.e., the summation and difference models, which is fully evaluated in this section. To do that, OTO(Linear) and

Table 4: Comparative results between OTO and its two variants on LIVE1

LIVE1	Algorithm	PSNR	PSNR-B	SSIM
QF-10	**JPEG**	27.77	25.33	0.7905
	OTO(Linear)	29.26	28.91	0.8295
	OTO(Sum)	29.26	28.91	0.8287
	OTO	**29.28**	**28.95**	**0.8298**
QF-20	**JPEG**	30.07	27.57	0.8683
	OTO(Linear)	31.62	31.12	0.8950
	OTO(Sum)	31.60	31.07	0.8941
	OTO	**31.67**	**31.17**	**0.8954**
QF-30	**JPEG**	31.41	28.92	0.9000
	OTO(Linear)	33.06	32.45	0.9218
	OTO(Sum)	32.65	32.27	0.9160
	OTO	**33.08**	**32.48**	**0.9218**
QF-40	**JPEG**	32.35	29.96	0.9173
	OTO(Linear)	34.03	33.43	0.9349
	OTO(Sum)	33.68	33.35	0.9316
	OTO	**34.10**	**33.48**	**0.9362**

OTO(Sum) are fully investigated in our comparison. The former one learns a weight factor to balance the significance of two branch networks, which adaptively combine two CNNs to solve the CAR problem. It is verified to be very effective for the reason that the significance of each CNN should be well considered in the fusion process. For the OTO(Sum) network, the weight factor is fixed to 1, which means that this version of OTO is not only shortage of the nonlinear representation ability but also impossible to tell which branch network contains more important information to suppress compression artifacts. In other words, OTO(Sum) just directly applies the addition operation to the two sub-networks. In this experiment, all comparative networks are in fair comparison, which are trained based on BSD500 training and testing sets, and then tested on LIVE1 and Classic5. The results are shown in Table 4 and Table 5. The OTO network along with its two variants have remarkable restoration performances on the four quality factors. These results demonstrate the effectiveness of the auto-learned weight factor α and the nonlinear operation on the summation and difference of the two branch networks. PSNR-B metric is designed specifically to measure the blocking artifacts. Particularly, we analyze the PSNR-B gain of OTO and OTO(Linear) compared to their baseline OTO(Sum). We observe that for low-quality (QF-10, QF-20) compression images, the nonlinear operation benefit more than the weight factor on suppressing blocking artifacts, but different for high-quality images (QF-30, QF-40). We trained OTO models on GTX 1070, I7-6700k with 32G memory. The training time for OTO, OTO(Linear), and OTO(Sum) are 6h12m, 5h42m and 5h31m, respectively. The average test time for OTO, OTO(Linear), and OTO(Sum) are 0.1803s, 0.1738s and 0.1734s per image, respectively.

We further evaluate how the weight factor α shown in Eq. 9 affect on the final performance. Results are shown in Table 6. α is implemented based on a scale layer of the Caffe platform, which can be updated by the BP algorithm. We can also give a constant α by manually setting the learning rate of this layer to be 0, so that α keeps unchanged during the training process. Firstly, we revisit OTO(Linear) and get the learned α, 0.0651 and 0.0544 for QF=20 and 30 respectively. The weight of the small-scaled sub-network is 20 times smaller than that of the normal-scaled sub-network, indicating that the normal-scaled features contain much richer information than small-scale ones. Then, we set α to 0.01, 0.1 and 1.0 (when $\alpha = 1.0$, it leads to OTO(Sum)). The results show that when α is set to 0.01 and 0.1, close to the auto-learned value, the performance is slightly worse than OTO(Linear)(learned α), but much better than OTO(Sum) ($\alpha = 1$), particularly on QF=30. Considering on all cases that OTO(Linear) achieves best results, we can conclude that

18

Table 5: Comparisons between OTO and its two variants on Classic 5

Classic5	Algorithm	PSNR	PSNR-B	SSIM
QF-10	JPEG	27.82	25.21	0.7800
	OTO(Sum)	29.36	28.92	0.8207
	OTO(Linear)	29.34	28.93	**0.8222**
	OTO	**29.36**	**28.94**	0.8222
QF-20	JPEG	30.12	27.50	0.8541
	OTO(Sum)	31.56	31.00	0.8767
	OTO(Linear)	31.54	31.01	0.8774
	OTO	**31.64**	**31.10**	**0.8785**
QF-30	JPEG	31.48	28.94	0.8844
	OTO(Sum)	32.52	31.99	0.8966
	OTO(Linear)	32.93	32.28	0.9021
	OTO	**32.95**	**32.33**	**0.9022**
QF-40	JPEG	32.43	29.92	0.9011
	OTO(Sum)	33.46	32.91	0.9114
	OTO(Linear)	33.77	33.06	0.9139
	OTO	**33.85**	**33.13**	**0.9155**

Table 6: The weight factor α evaluation experiments on LIVE1

LIVE1	α	PSNR	PSNR-B	SSIM
QF-20	0.01	31.59	31.12	0.8947
	0.1	31.59	31.12	0.8949
	1.0	31.60	31.07	0.8941
	0.0651(auto-learned)	**31.62**	**31.12**	**0.8950**
QF-30	0.01	33.03	32.41	0.9214
	0.1	33.04	32.43	0.9214
	1.0	32.65	32.27	0.9160
	0.0544(auto-learned)	**33.06**	**32.45**	**0.9218**

an auto-learned α is significant for a practical CAR system especially when a proper α cannot be given in advance. In addition, OTO(Sum) means that no difference model is in use, in contrast our OTO with the difference model always achieve a better performance as shown in Table 4 and Table 5, which prove that OTOs benefit from the high frequency information. We visualize the feature maps of the summation model and the difference models in Fig. 6 for a picture from LIVE1, and the results clearly support our motivation.

4.4. On Remote Sensing Image Datasets

Figure 7: The difference between JPEG and SPIHT compression algorithm. Left: JPEG with block size 8×8, Right: SPIHT with block size 32×32. The blocking artifact caused by SPIHT is more severe than by JPEG.

For JPEG-based compression artifacts reduction methods, their target block size is 8×8, but in our SPIHT-based algorithm, blocking artifacts with a larger size like 32×32 will occur, which is shown in Fig. 7. Remotely sensed images are quite different from the natural images like BSD500 in terms of color richness, texture distribution and so on. ARCNN is first designed for restoring natural images. For a fair comparison, we retrain ARCNN on the remote sensed image dataset with the architecture of the network unchanged. We adopt better training parameters with step-attenuated learning rate compared to its fixed one. The network tends to converge early in the four compression rates, 8, 16, 32 and 64 and then we evaluate it on the remote sensing task. We train and test our OTOs on remotely sensed image dataset, and the results are shown in Table 7. The

20

Table 7: Results on Remotely Sensed Dataset

Quality	Evaluation	SPIHT	ARCNN	OTO
8	PSNR	37.19	35.71	**39.21**
	SSIM	0.9782	0.9775	**0.9854**
	PSNR-B	31.90	33.12	**37.95**
16	PSNR	33.47	32.69	**35.23**
	SSIM	0.9523	0.9554	**0.9652**
	PSNR-B	28.80	30.61	**34.38**
32	PSNR	30.47	30.39	**32.23**
	SSIM	0.9111	0.9200	**0.9361**
	PSNR-B	26.31	28.54	**31.53**
64	PSNR	27.90	27.83	**29.54**
	SSIM	0.8489	0.8547	**0.8875**
	PSNR-B	24.19	26.55	**29.20**

parameters in PSNR-B and SSIM algorithms are modified to evaluate the 32×32-sized blocking artifacts.

It is astonishing that in various compression rates, ARCNN does not increase three scores except for PSNR-B, while OTO successfully suppressed compression artifacts on all measures. In Fig. 10 and Fig. 11, the images restored by ARCNN tend to be blurry with blocking artifacts remained. The failure of ARCNN and the success of the OTO verify that OTO are quite effective for remote sensing images restoration, when suffering from larger blocking artifacts problems. However, when compression rate becomes bigger, i.e., 64, the details of the compressed images are almost lost, our OTO fail to restore the edges and structure details of the balcony as shown in Fig. 12.

4.5. On LIVE1 and BSD500 Tests sets

LIVE1: As mentioned above, the proposed OTO outperform ARCNN on the remote sensing image dataset and shows the promising results on restoring SPIHT-based compression artifacts. The following experiments further support that even compared with recently proposed deep learning methods, OTO can still achieve the state-of-the-art results on publicly LIVE1 and BSD500 test sets based on the JPEG compression.

We compare OTO with the most successful deblocking oriented method, SA-DCT, which achieves the state-of-the-art results. Then ARCNN is also included for a complete assessment, using the

Table 8: Results on LIVE1

LIVE1	Algorithm	PSNR	PSNR-B	SSIM
	JPEG	27.77	25.33	0.7905
QF-10	**SA-DCT**	28.65	28.01	0.8093
	AR-CNN	29.13	28.74	0.8232
	OTO	**29.28**	**28.95**	**0.8298**
	JPEG	30.07	27.57	0.8683
QF-20	**SA-DCT**	30.81	29.82	0.8781
	AR-CNN	31.40	30.69	0.8886
	OTO	**31.67**	**31.17**	**0.8954**
	JPEG	31.41	28.92	0.9000
QF-30	**SA-DCT**	32.08	30.92	0.9078
	AR-CNN	32.69	32.15	0.9166
	OTO	**33.08**	**32.48**	**0.9218**
	JPEG	32.35	29.96	0.9173
QF-40	**SA-DCT**	32.99	31.79	0.9240
	AR-CNN	33.63	33.12	0.9306
	OTO	**34.10**	**33.48**	**0.9362**

same metric as before. The results are shown in Table 8. ARCNN does not use data augmentation technique on the training set in the initial conference version, but in its extended journal version $20\times$ augmentation method is used so as to gain restoration performance improvement. In our experiments, no data augmentation is applied with the aim to accelerate the training process. Specifically, for the PSNR metric, we achieve an average gain of 0.90 dB compared with SA-DCT and 0.32 dB compared with ARCNN. For the PSNR-B metric, the gains are even larger to 1.38 dB and 0.34 dB respectively. It shows that OTOs are suitable for suppressing compression artifacts for natural images.

BSD500: We compare OTO with the traditional approaches like DSC and also convolutional deep learning based approaches, such as ARCNN and Trainable Nonlinear Reaction Diffusion (TNRD) [60]. In DDCN, the DCT-Domain branch took advantage of JPEG-based prior so it is unfair for OTO only using pixel-domain information. Guo et al. propose a variant of DDCN by removing the DCT-domain branch so that no extra prior is utilized, which is alternatively used in the comparison. The comparative results are shown in Table 9 with four quality factors from 10

Table 9: Results on BSD500 Test Set

Quality	Evaluation	JPEG	DSC	DDCN(-DCT)	TNRD	ARCNN	OTO
10	PSNR	27.8	28.79	29.26	29.16	29.10	**29.31**
	SSIM	0.7875	0.8124	0.8267	0.8225	0.8198	**0.8278**
	PSNR-B	25.1	28.45	28.89	28.81	28.73	**28.92**
20	PSNR	30.05	30.97	31.55	31.41	31.28	**31.64**
	SSIM	0.8671	0.8804	0.8923	0.8889	0.8854	**0.8943**
	PSNR-B	27.22	30.57	30.84	30.83	30.55	**30.95**
30	PSNR	31.37	32.29	32.92	32.77	32.67	**33.03**
	SSIM	0.8994	0.9093	0.9193	0.9166	0.9152	**0.9211**
	PSNR-B	28.53	31.84	32.01	31.99	31.94	**32.17**
40	PSNR	32.3	33.23	33.87	33.73	33.55	**34.00**
	SSIM	0.9171	0.9253	0.9336	0.9316	0.9296	**0.9357**
	PSNR-B	29.49	32.71	32.86	32.79	32.78	**32.98**

to 40. OTO outperforms all the other algorithms in terms of three metrics, which indicates that OTO has a competent restoration ability. More specifically, OTO obtains about 0.7 dB and 0.4 dB gains compared with DSC on the PSNR and PSNR-B respectively. ARCNN is beaten by 0.35 dB on the PSNR and 0.26 dB on the PSNR-B, which is consistent with the results on LIVE1.

Original
PSNR/SSIM/PSNR-B

JPEG
32.92/0.9316/29.67

ARCNN
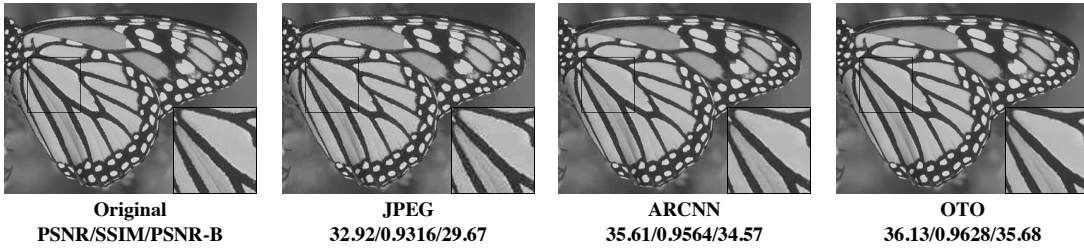**35.61/0.9564/34.57**

OTO
36.13/0.9628/35.68

Figure 8: Qualitative comparison of OTO and ARCNN by JPEG with Quality Factor=20 where ringing effects is carefully handled after being restored by OTO network.

5. Conclusion and future work

The CAR problem is a challenging in the field of remote sensing. In this paper, we have developed a new and general framework to combine different models based on a non-linear method

23

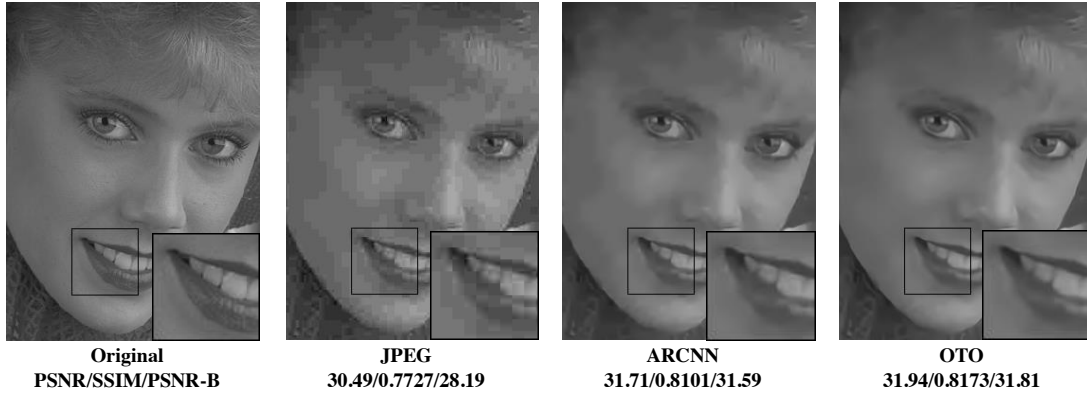| **Original** | **JPEG** | **ARCNN** | **OTO** |
| **PSNR/SSIM/PSNR-B** | **30.49/0.7727/28.19** | **31.71/0.8101/31.59** | **31.94/0.8173/31.81** |

Figure 9: Qualitative comparison of OTO and ARCNN by JPEG with Quality Factor=10, where severe block artifacts are removed and the edges are sharp again.
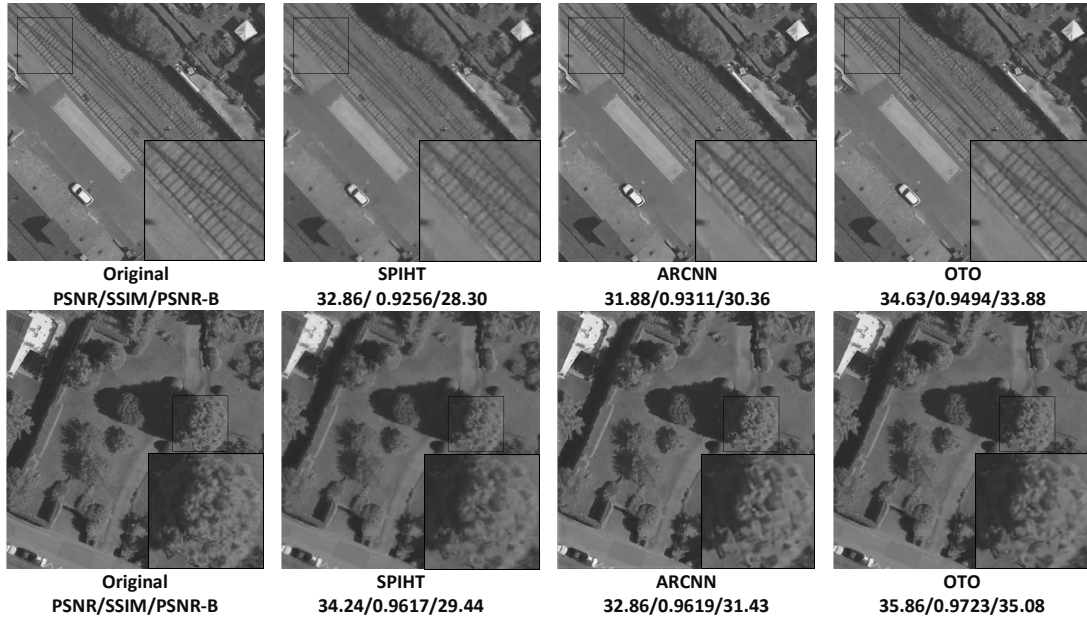

Original	**SPIHT**	**ARCNN**	**OTO**
PSNR/SSIM/PSNR-B	**32.86/ 0.9256/28.30**	**31.88/0.9311/30.36**	**34.63/0.9494/33.88**
Original	**SPIHT**	**ARCNN**	**OTO**
PSNR/SSIM/PSNR-B	**34.24/0.9617/29.44**	**32.86/0.9619/31.43**	**35.86/0.9723/35.08**

Figure 10: Qualitative comparison of OTO and ARCNN by SPIHT with Compression Rate=16.

to effectively deal with complicated compression artifacts, i.e., big blocking effect in the compression. Based on the Taylor expansion, we lead to two simple OTO variants, which provide a more profound investigation into our method and pose a new direction to solve the artifact reduction problem. Extensive experiments are conducted to validate the performance of OTO and new state-of-the-art

24

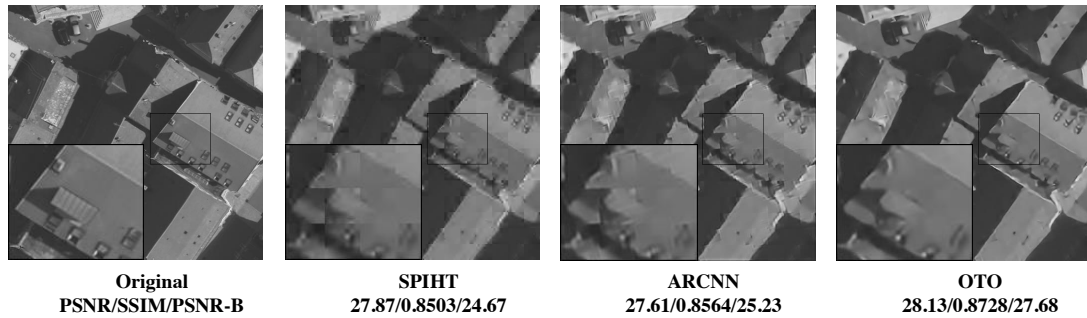Figure 11: Qualitative comparison of OTO and ARCNN by SPIHT with Compression Rate=32.

Figure 12: Qualitative comparison of OTO and ARCNN by SPIHT with Compression Rate=64.

results are obtained. In the future work, we will deploy more complicated network in our framework to gain better performance.

References

[1] C. Chen, B. Zhang, H. Su, W. Li, L. Wang, Land-use scene classification using multi-scale completed local binary patterns, Signal, Image and Video Processing 10 (4) (2016) 745–752.

[2] J. Li, P. Gamba, A. Plaza, A novel semi-supervised method for obtaining finer resolution urban extents exploiting coarser resolution maps, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 7 (10) (2014) 4276–4287.

[3] W. Li, C. Chen, H. Su, Q. Du, Local binary patterns and extreme learning machine for hyperspectral imagery classification, IEEE Transactions on Geoscience and Remote Sensing 53 (7) (2015) 3681–3693.

[4] G. Vosselman, M. Coenen, F. Rottensteiner, Contextual segment-based classification of airborne laser scanner data, ISPRS Journal of Photogrammetry and Remote Sensing 128 (2017) 354–371.

[5] Z. Wang, A. C. Bovik, L. Lu, Why is image quality assessment so difficult?, in: Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on, Vol. 4, IEEE, 2002, pp. 4–3313.

[6] K. Yu, C. Dong, C. C. Loy, X. Tang, Deep convolution networks for compression artifacts reduction, arXiv preprint arXiv:1608.02778.

[7] A. Said, W. A. Pearlman, A new, fast, and efficient image codec based on set partitioning in hierarchical trees, IEEE Transactions on circuits and systems for video technology 6 (3) (1996) 243–250.

[8] G. Cheng, J. Han, A survey on object detection in optical remote sensing images, ISPRS Journal of Photogrammetry and Remote Sensing 117 (2016) 11 – 28.

[9] P. Xiao, X. Zhang, D. Wang, M. Yuan, X. Feng, M. Kelly, Change detection of built-up land: A framework of combining pixel-based detection and object-based recognition, ISPRS Journal of Photogrammetry and Remote Sensing 119 (2016) 402–414.

[10] X. Bian, C. Chen, L. Tian, Q. Du, Fusing local and global features for high-resolution scene classification, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 10 (6) (2017) 2889–2901.

[11] C.-I. Chang, S.-S. Chiang, Anomaly detection and classification for hyperspectral imagery, IEEE Transactions on Geoscience and Remote Sensing 40 (6) (2002) 1314–1325.

[12] P. List, A. Joch, J. Lainema, G. Bjontegaard, M. Karczewicz, Adaptive deblocking filter, IEEE transactions on circuits and systems for video technology 13 (7) (2003) 614–619.

[13] H. C. Reeve III, J. S. Lim, Reduction of blocking effects in image coding, Optical Engineering 23 (1) (1984) 230134–230134.

[14] C. Wang, J. Zhou, S. Liu, Adaptive non-local means filter for image deblocking, Signal Processing: Image Communication 28 (5) (2013) 522–530.

[15] X. Liu, X. Wu, J. Zhou, D. Zhao, Data-driven sparsity-based restoration of jpeg-compressed images in dual transform-pixel domain, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5171–5178.

[16] A.-C. Liew, H. Yan, Blocking artifacts suppression in block-coded images using overcomplete wavelet representation, IEEE Transactions on Circuits and Systems for Video Technology 14 (4) (2004) 450–461.

[17] A. Foi, V. Katkovnik, K. Egiazarian, Pointwise shape-adaptive dct for high-quality denoising and deblocking of grayscale and color images, IEEE Transactions on Image Processing 16 (5) (2007) 1395–1411.

[18] C. Dong, C. C. Loy, K. He, X. Tang, Learning a deep convolutional network for image super-resolution, in: European Conference on Computer Vision, Springer, 2014, pp. 184–199.

[19] L. Cavigelli, P. Hager, L. Benini, Cas-cnn: A deep convolutional neural network for image compression artifact suppression, in: Neural Networks (IJCNN), 2017 International Joint Conference on, IEEE, 2017, pp. 752–759.

[20] K. Zhang, Y. Chen, Y. Chen, D. Meng, L. Zhang, Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising, IEEE Transactions on Image Processing 26 (7) (2017) 3142–3155.

[21] D. S. Early, D. G. Long, Image reconstruction and enhanced resolution imaging from irregular samples, IEEE Transactions on Geoscience and Remote Sensing 39 (2) (2001) 291–302.

[22] G. Vivone, L. Alparone, J. Chanussot, M. Dalla Mura, A. Garzelli, G. A. Licciardi, R. Restaino, L. Wald, A critical comparison among pansharpening algorithms, in: Geoscience and Remote Sensing Symposium, 2015, pp. 2565–2586.

27

[23] J. Kim, J. Kwon Lee, K. Mu Lee, Accurate image super-resolution using very deep convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1646–1654.

[24] T. Wang, M. Sun, K. Hu, Dilated deep residual network for image denoising, arXiv preprint arXiv:1708.05473.

[25] B. Lim, S. Son, H. Kim, S. Nah, K. M. Lee, Enhanced deep residual networks for single image super-resolution, in: Computer Vision and Pattern Recognition Workshops, 2017, pp. 1132–1140.

[26] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, Photo-realistic single image super-resolution using a generative adversarial network, arXiv preprint arXiv:1609.04802.

[27] G. Masi, D. Cozzolino, L. Verdoliva, G. Scarpa, Cnn-based pansharpening of multi-resolution remote-sensing images, in: Urban Remote Sensing Event, 2017, pp. 1–4.

[28] G. Scarpa, S. Vitale, D. Cozzolino, Target-adaptive cnn-based pansharpening, arXiv preprint arXiv:1709.06054.

[29] K. Zhang, W. Zuo, L. Zhang, Ffdnet: Toward a fast and flexible solution for cnn based image denoising, arXiv preprint arXiv:1710.0402.

[30] S. Paris, S. W. Hasinoff, J. Kautz, Local laplacian filters: Edge-aware image processing with a laplacian pyramid., ACM Trans. Graph. 30 (4) (2011) 68–1.

[31] P. Burt, E. Adelson, The laplacian pyramid as a compact image code, IEEE Transactions on communications 31 (4) (1983) 532–540.

[32] D. J. Heeger, J. R. Bergen, Pyramid-based texture analysis/synthesis, in: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, ACM, 1995, pp. 229–238.

[33] K. Sunkavalli, M. K. Johnson, W. Matusik, H. Pfister, Multi-scale image harmonization, in: ACM Transactions on Graphics (TOG), Vol. 29, ACM, 2010, p. 125.

[34] J. Jancsary, S. Nowozin, C. Rother, Loss-specific training of non-parametric image restoration models: A new state of the art, in: European Conference on Computer Vision, Springer, 2012, pp. 112–125.

[35] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images, Technical report, University of Toronto, 2009.

[36] G. Cheng, J. Han, X. Lu, Remote sensing image scene classification: Benchmark and state of the art, Proceedings of the IEEE 105 (2017) 1865–1883.

[37] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in neural information processing systems, 2015, pp. 91–99.

[38] X. Yao, J. Han, L. Guo, S. Bu, Z. Liu, A coarse-to-fine model for airport detection from remote sensing images using target-oriented visual saliency and crf, Neurocomputing 164 (2015) 162–172.

[39] J. Han, D. Zhang, G. Cheng, L. Guo, J. Ren, Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning, IEEE Transactions on Geoscience and Remote Sensing 53 (6) (2015) 3325–3337.

[40] G. Cheng, P. Zhou, J. Han, Learning rotation-invariant convolutional neural networks for object detection in vhr optical remote sensing images, IEEE Transactions on Geoscience and Remote Sensing 54 (12) (2016) 7405–7415.

[41] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.

[42] X. Yao, J. Han, D. Zhang, F. Nie, Revisiting co-saliency detection: A novel approach based on two-stage multi-view spectral rotation co-clustering, IEEE Transactions on Image Processing 26 (7) (2017) 3196–3209.

[43] X. Yao, J. Han, G. Cheng, X. Qian, L. Guo, Semantic annotation of high-resolution satellite images via weakly supervised learning, IEEE Transactions on Geoscience and Remote Sensing 54 (6) (2016) 3660–3671.

[44] R. Timofte, V. De Smet, L. Van Gool, A+: Adjusted anchored neighborhood regression for fast super-resolution, in: Asian Conference on Computer Vision, Springer, 2014, pp. 111–126.

[45] P. Svoboda, M. Hradis, D. Barina, P. Zemcik, Compression artifacts removal using convolutional neural networks, arXiv preprint arXiv:1605.00366.

[46] G. Leonardo, S. Lorenzo, B. Marco, A. D. Bimbo, Deep generative adversarial compression artifact removal, arXiv preprint arXiv:1704.02518.

[47] J. Guo, H. Chao, Building dual-domain representations for compression artifacts reduction, in: European Conference on Computer Vision, Springer, 2016, pp. 628–644.

[48] Z. Wang, D. Liu, S. Chang, Q. Ling, Y. Yang, T. S. Huang, D3: Deep dual-domain based fast restoration of jpeg-compressed images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2764–2772.

[49] K. Gregor, Y. LeCun, Learning fast approximations of sparse coding, in: Proceedings of the 27th International Conference on Machine Learning (ICML-10), 2010, pp. 399–406.

[50] G. Jun, C. Hongyang, One-to-many network for visually pleasing compression artifacts reduction, arXiv preprint arXiv:1611.04994.

[51] K. He, X. Zhang, S. Ren, J. Sun, Identity mappings in deep residual networks, in: European Conference on Computer Vision, Springer, 2016, pp. 630–645.

[52] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, arXiv preprint arXiv:1502.03167.

[53] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.

[54] G. Huang, Z. Liu, K. Q. Weinberger, L. van der Maaten, Densely connected convolutional networks, arXiv preprint arXiv:1608.06993.

[55] P. Arbelaez, M. Maire, C. Fowlkes, J. Malik, Contour detection and hierarchical image segmentation, IEEE transactions on pattern analysis and machine intelligence 33 (5) (2011) 898–916.

[56] H. Sheikh, Z. Wang, L. Cormack, A. Bovik, Live image quality assessment database release 2 (2005).

[57] M. Cramer, The dgpf-test on digital airborne camera evaluation–overview and test design, Photogrammetrie-Fernerkundung-Geoinformation 2010 (2) (2010) 73–82.

[58] C. Yim, A. C. Bovik, Quality assessment of deblocked images, IEEE Transactions on Image Processing 20 (1) (2011) 88–98.

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

[60] Y. Chen, W. Yu, T. Pock, On learning optimized reaction diffusion processes for effective image restoration, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5261–5269.

Biography

Baochang Zhang. received the B.S., M.S. and Ph.D. degrees in Computer Science from Harbin Institute of the Technology, Harbin, China, in 1999, 2001, and 2006, respectively. From 2006 to 2008, he was a research fellow with the Chinese University of Hong Kong, Hong Kong, and with Griffith University, Brisban, Australia. Currently, he is an associate professor with the Science and Technology on Aircraft Control Laboratory, School of Automation Science and Electrical Engineering, Beihang University, Beijing, China. He was supported by the Program for New Century Excellent Talents in University of Ministry of Education of China. His current research interests include pattern recognition, machine learning, face recognition, and wavelets.

Jiaxin Gu. received the B.S. degree in School of Automation Science and Electrical Engineering of Beihang University in 2017. He is pursuing his Master degree in the same shool of Beihang University and his current research interests include image restoration, object detection and deep learning.

Chen Chen. received the B.E. degree in automation from the Beijing Forestry University, Beijing, China, in 2009, and the M.S. degree in electrical engineering from the Mississippi State University, Starkville, MS, USA, in 2012, and the Ph.D. degree in the Department of Electrical Engineering, University of Texas at Dallas, Richardson, TX, USA, in 2016. He is currently a Post-Doc in the Center for Research in Computer Vision, University of Central Florida, Orlando, FL, USA. He has published more than 50 papers in refereed journals and conferences in these areas. His research interests include compressed sensing, signal and image processing, pattern recognition, and computer vision.

Jungong Han. is currently a tenured faculty member with the School of Computing and Communications at Lancaster University, UK. His research interestes include video analysis, computer vision and artificial intelligence.

Xiangbo Su. received his B.E. degree in Automation Science from Beihang University, Beijing, China, in 2015. He is currently a M.S. student at the School of Automation Science and Electrical Engineering, Beihang University. His current research interests include machine learning and deep learning in general, with computer vision applications in object tracking, recognition and image restoration.

Xianbin Cao. received the B.Eng and M.Eng degrees in computer applications and information science from Anhui University, Hefei, China, in 1990 and 1993, respectively, and the Ph.D. degree in information science from the University of Science and Technology of China, Beijing, in 1996. He is currently a Professor with the School of Electronic and Information Engineering, Beihang University, Beijing, China. He is also the Director of the Laboratory of Intelligent Transportation System. His current research interests include intelligent transportation systems, airspace transportation management, and intelligent computation.

Jianzhuang Liu. received the Ph.D. degree in computer vision from The Chinese University of Hong Kong, Hong Kong, in 1997. He was a Research Fellow with Nanyang Technological University, Singapore, from 1998 to 2000. From 2000 to 2012, he was a Post-Doctoral Fellow, an Assistant Professor, and an Adjunct Associate Professor with The Chinese University of Hong Kong. He was a Professor in 2011 with the University of Chinese Academy of Sciences. He is currently a Principal Researcher with Huawei Technologies Company, Ltd., Shenzhen. He has authored over 100 papers,

most of which are in prestigious journals and conferences in computer science. His research interests include computer vision, image processing, machine learning, multimedia, and graphics.