Reference-Based Multi-Stage Progressive Restoration for Multi-Degraded Images

Yi Zhang, Qixue Yang, Damon M. Chandler, *Senior Member, IEEE*, and Xuanqin Mou, *Senior Member, IEEE*

***Abstract*— Image restoration (IR) via deep learning has been vigorously studied in recent years. However, due to the ill-posed nature of the problem, it is challenging to recover the high-quality image details from a single distorted input especially when images are corrupted by multiple distortions. In this paper, we propose a multi-stage IR approach for progressive restoration of multi-degraded images via transferring similar edges/textures from the reference image. Our method, called a Reference-based Image Restoration Transformer (Ref-IRT), operates via three main stages. In the first stage, a cascaded U-Transformer network is employed to perform the preliminary recovery of the image. The proposed network consists of two U-Transformer architectures connected by feature fusion of the encoders and decoders, and the residual image is estimated by each U-Transformer in an easy-to-hard and coarse-to-fine fashion to gradually recover the high-quality image. The second and third stages perform texture transfer from a reference image to the preliminarily-recovered target image to further enhance the restoration performance. To this end, a quality-degradation-restoration method is proposed for more accurate content/texture matching between the reference and target images, and a texture transfer/reconstruction network is employed to map the transferred features to the high-quality image. Experimental results tested on three benchmark datasets demonstrate the effectiveness of our model as compared with other state-of-the-art multi-degraded IR methods. Our code and dataset are available at https://vinelab.jp/refmdir/.**

***Index Terms*— Image restoration, U-transformer, transposed attention, texture transfer.**

I. Introduction

A. Background

IMAGE restoration (IR) which aims to recover a clean image from its degraded version is a classic and fundamental problem in image processing and computer vision. Typical examples of image degradation include noise, blur, ringing, blocking, rain, snow, haze, etc. These degradations not only harm the quality of the user experience, but also have a negative impact on various computer vision applications that take the degraded images as input. Thus, IR has been extensively studied over the last several decades, and numerous IR techniques have been proposed.

Since IR is typically an ill-posed inverse problem that has infinite feasible solutions, previous model-based methods often employ image priors to restrict the solution space to that of the optimal high-quality natural images. Successful image priors include the total variation prior (e.g., [1]), sparse representation prior (e.g., [2], [3]), low-rank prior (e.g., [4], [5]), and non-local self-similarity prior (e.g., [6], [7]), etc. Some approaches (e.g., [8], [9], [10], etc.) employ more than one image prior for effective restoration. However, designing such handcrafted priors is non-trivial, and one image prior that works for a certain distortion type may not be able to work for others. In comparison, convolutional neural network (CNN) based methods that learn more general priors from large-scale data have gained attention in recent years, and have now become the mainstream solution to IR.

One way to improve the performance of the CNN-based IR methods is to design more powerful and effective network models. Thus, considerable effort has been made to optimize the network architecture. Starting from the first image super-resolution network (i.e., SRCNN [11]) that consists of only three convolution layers, numerous network modules and functional units have been developed including residual learning [12], dilated/deformable convolution [13], [14], dense connection [15], encoder-decoders [16], generative adversarial models [17], recurrent/recursive network [18], [19], etc. Though effective, the CNN-based model still suffers from two important issues. One is the limited receptive field due to the fixed convolution kernel size, and the other one is the weak adaptiveness and flexibility to different image contents due to the static kernel weights at inference. Both issues ultimately prevent further improvement of the CNN-based methods in solving the IR problem.

To overcome these limitations, the non-local operation [20] and self-attention (SA) mechanism [21] were developed, both of which allow distant pixels to contribute to the response at a given position based on pixel/patch similarity. Although the two methods are highly effective in capturing long-range pixel interactions, their computational complexities grow quadratically with image size, which prevent their wider applications on high-resolution images. To reduce the computational cost of SA, the window partition mechanism was introduced, leading to the Swin Transformer [22] architecture which has been widely used in many computer vision tasks such as image classification [23], [24], detection [25], [26], fusion [27], [28], segmentation [29], [30], and restoration [31]. Meanwhile, the transposed attention [32] was developed which applies SA across the channel dimension rather than the spatial

Manuscript received 11 February 2023; revised 9 July 2024 and 14 August 2024; accepted 23 August 2024. Date of publication 5 September 2024; date of current version 11 September 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 61901355, Grant 62071375, and Grant 62271384; and in part by the Japan Society for the Promotion of Science under Grant 22K12085. The associate editor coordinating the review of this article and approving it for publication was Dr. Lei Zhang. *(Corresponding author: Yi Zhang.)*

Yi Zhang, Qixue Yang, and Xuanqin Mou are with the School of Information and Communications Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: yi.zhang.osu@xjtu.edu.cn).

Damon M. Chandler is with the College of Information Science and Engineering, Ritsumeikan University, Osaka 567-8570, Japan.

Digital Object Identifier 10.1109/TIP.2024.3451939

Fig. 1. An example of applying the retrained Restormer [33] model vs. our method on a multi-degraded image generated from one of the pristine images (006_0.png) in the CUFED5 dataset [34]. (a) Distorted image; (b) reference image; (c) restored image output by Restormer; (d) restored image output by the proposed method.

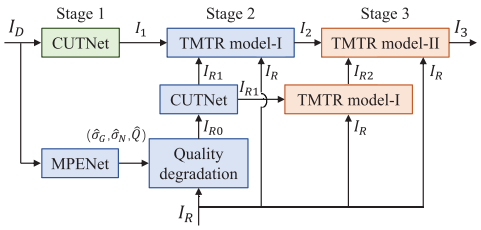


Fig. 2. A block diagram of the proposed Ref-IRT model. Note that the distorted image I_D is progressively restored via three main stages. The first stage performs preliminary restoration; the second primary restoration stage improves the IR performance based on edge/texture matching and transferring between the reference and distorted images; and the third stage performs the final restoration to further enhance the restored image quality.

dimension to exploit the interdependencies between channel maps. Recently, the convolution layers in U-Net [16] were replaced by the multi-Dconv head transposed attention modules, resulting in an efficient Transformer model (i.e., Restormer [33]) for high-resolution IR.

B. Motivation

Despite the development of these various modules for IR, there remains one major limitation: most existing methods were only designed and trained to deal with images contaminated by a single distortion type at a fixed distortion level. Because images can be degraded by many unknown factors during the image acquisition, compression, transmission, reception, and display stages, an effective multi-degraded image restoration (MDIR) method is desirable. Although some research toward the MDIR problem is beginning to be reported (e.g., [35], [36], [37], [38], [39]), their performances are relatively weak due partly to the pure CNN-based architectures being adopted. One potential solution is to train the advanced single-image IR networks (e.g., [33]) that mitigate the shortcomings of CNN via a Transformer-based architecture on multi-degraded images to obtain a decent MDIR model (a strategy also adopted in this work). However, as we have found, these retrained models do not perform sufficiently well [as shown in Figure 1(c)], which might be attributed to the fact that single-image IR has reached its upper-bound performance on multi-degraded images even with the most advanced deep learning techniques due to the irreversible nature of the image degradation process.

Motivated by the recent image super-resolution (SR) works (e.g., [40], [41], [42], [43]) that utilize an external reference image to help super-resolve the low-resolution image, it is reasonable to believe that the perceived quality of a multi-degraded image can also be enhanced if a reference image with similar content is given. In this way, the side information from the reference image can be transferred to the distorted image to help restore the edge/texture/structure information lost in the degradation process. Of course, a key requirement of this approach is the ability to perform accurate content/texture matching between the distorted and reference images. For image SR, the similar content can be found by simply downsampling and upsampling the high-resolution reference image before the dense patch matching is applied. However, as demonstrated later in Table I, this method cannot be used for restoration of multi-degraded images because distortion artifacts can make the matching process much more difficult even if the two images share a high degree of similarity [as shown in Figure 1 (a) and (b)]. Thus, in this paper, we propose a three-stage framework for reference-based MDIR; the main idea is to perform dense patch matching on two moderate-quality images with similar distortion artifacts: one image is the preliminarily/primarily-recovered distorted image and the other image is the reference image that has been subjected to the same quality-degradation-restoration process. As demonstrated in Figure 1 (d), the proposed reference-based IR method can add more details to the edges/textures of the image and can thus significantly improve the perceived quality.

C. Proposal and Contributions

Based on the abovementioned points, in this paper, we propose a new MDIR model called a Reference-based Image Restoration Transformer (Ref-IRT) to restore images simultaneously corrupted by three distortion types: Gaussian blur, white noise, and JPEG compression. As shown in Figure 2, Ref-IRT operates via three main stages:

The first stage (marked by green) employs a cascaded U-Transformer network (CUTNet) to perform the preliminary restoration. Unlike [33] that employs only one U-Transformer network to predict the residual, our model consists of two U-Transformer modules, which sequentially predict the residual in an easy-to-hard and coarse-to-fine manner.

The second stage (marked by blue) performs the primary restoration by transferring similar edges/textures from the reference image I_R to the preliminarily-recovered image I_1 to further improve its perceived quality. To this end, a multiple-distortion parameter estimation network (MPENet) was developed to predict the three distortion parameters ($\hat{\sigma}_G$, $\hat{\sigma}_N$, $\hat{Q}$), which are used to add the same amounts of the three distortions to the reference image. Then, the quality-degraded reference image I_{R0} is processed by the same CUTNet to obtain a quality-restored reference image I_{R1}. Subsequently, I_1, I_{R1}, and I_R are fed into the texture matching, transfer, and reconstruction (TMTR) model-I to obtain the primarily-recovered image I_2.

The third stage (marked by orange) performs the final restoration by transferring more similar edges/textures from I_R to I_2. To this end, I_{R1} is first processed by the TMTR model-I to obtain the quality-improved reference image I_{R2}, which, along with I_2 and I_R, are then fed into the TMTR model-II to obtain the final output I_3. Since I_2 and I_{R2} are of better quality than I_1 and I_{R1}, more accurate patch matching can be made between I_2 and I_{R2}, which ultimately results in a quality-enhanced image I_3.

The main contributions of this work are as follows:

1) We present a reference-based MDIR model which for the first time utilizes an external reference image to help restore the perceived quality of multi-degraded images. To this end, we propose a three-stage IR pipeline that

can be theoretically generalized to images corrupted by any distortion types and at any distortion levels.

2) We propose a cascaded U-Transformer network to perform restoration on multi-degraded images. Different from existing MDIR works, our model progressively estimates the residual image in an easy-to-hard and coarse-to-fine manner, which allows less weight being assigned to the easy/coarse prediction network and more weight to the hard/fine prediction network. As a result, better restoration performance is realized with a relatively smaller number of network parameters.
3) We design a texture matching, transfer and reconstruction network for further enhancement of the restored image quality by using the edge/texture/structure information extracted from the reference image. To this end, we present a quality-degradation-restoration method for more accurate content/texture matching between the target distorted and reference images. Furthermore, we incorporate the dual-window mechanism within the transposed attention network to maximize the restoration capacity of the U-Transformer network.
4) We contribute a new XJTU-referenced image restoration (XRIR) dataset to evaluate the performance of the existing reference-based IR methods. The proposed dataset contains 200 high-resolution pristine images each of which has a corresponding reference. This dataset could also be used as a benchmark for performance evaluation of the other reference-based computer vision/image processing algorithms such as reference-based image super-resolution.

The rest of the paper is organized as follows. Section II reviews the existing MDIR methods and the reference-based IR approaches. Section III describes details of the proposed Ref-IRT model. In Section IV, we evaluate the performance of Ref-IRT on various multi-degraded image datasets. Experimental results show that with a comparable number of network parameters, floating point operations, and running time, Ref-IRT outperforms most existing MDIR methods in terms of four objective quality measures, and can possibly be extended to handle real-world distortions. General conclusions are presented in Section V.

II. Related Work

In this section, we provide a brief review of the existing MDIR methods and the reference-based IR methods, which are closely related to the proposed algorithm.

A. MDIR Methods

Most existing MDIR methods take the advantage of the power of deep learning, and attempt to build a deep CNN model which directly maps the multi-degraded image to the clean one. Different methods mainly differ in the network architectures being used and the degradation model being considered. Starting from the first RL-Restore model [44] which was designed to address the sequentially-applied multiple distortions, a number of MDIR networks have been proposed. For example, Suganuma et al. [36] presented an operation-wise attention network (OWAN); Huang et al. [38] presented a high-order form of OWAN (HOWAN); Liu et al. [35] presented a recurrent multi-branch network (RMBN). By considering different distortion types on different image areas, Kim et al. [37] presented a mixture of experts with a parameter sharing (MEPS) network, and Li et al. [45] presented the feature disentanglement and aggregation modules. Later, Shin et al. [39] presented the distortion information-guided network, which jointly considers the sequentially-applied and spatially-varying multiple distortions. In addition, to further take into account the diverse degradations in real-world images, Zhang et al. [46] proposed the BSRGAN framework which was trained on images corrupted by a practical degradation model. Although these MDIR methods are effective, the limits imposed by using only the distorted images and the CNN architectures suggest that their performances still have room for further enhancement especially when images are highly degraded.

B. Reference-Based IR Methods

To further improve the IR performance, other approaches have used reference images to help restore the target image. Based on the different restoration tasks being addressed, these reference-based IR methods mainly cover two branches: image super-resolution (SR) and inpainting.

For reference-based image SR, great efforts have been spent on (1) finding the accurate correspondence between the low-resolution (LR) and reference images, and (2) transferring the most relevant features from the reference image to the LR image to produce the visually favorable high-resolution (HR) image. For example, the attention mechanism was used in [40] which formulated the LR and reference images as queries and keys in a Transformer. In [47], the cross-scale warping was proposed to perform the non-rigid image transformation, and FlowNetS [48] was adopted to generate the multi-scale correspondence between the LR and HR images. In [49], the conditional variational auto-encoder was proposed to learn explicit distributions from the reference images. In [41], a coarse-to-fine correspondence matching scheme was proposed to speed up the matching process, and the distribution of the reference features was remapped in a spatially adaptive manner such that the algorithm is robust to reference images with different color and luminance distributions. In [42], the contrastive correspondence network and a teacher-student correlation distillation method were proposed to handle the transformation gap and the resolution gap, respectively. Recently, the deformable attention Transformer was proposed in [43] which extracted image transformation insensitive features for texture matching. Other similar works can be found in [34], [50], and [51], etc.

For reference-based image inpainting, the primary challenge is to precisely place the pixels from the reference image into the hole (missing) region in the target image. To address this challenge, Zhou et al. [52] proposed a multi-homography transformed fusion pipeline to obtain the multiple transformations of the source image, each of which aligns a specific region to the target image; Liu et al. [53] proposed a reference-based encoder-decoder network to jointly fill image holes, where the features of the target and reference images are aligned and fused by a feature alignment module; Zhao et al. [54] proposed a principled approach which fills the hole region by explicitly estimating the camera positions and geometry of the 3D scene from two limited views. Besides the aforementioned general image SR/inpainting algorithms, there are also some other literatures (e.g., [55], [56]) that utilized the reference settings for face image restoration/enhancement.

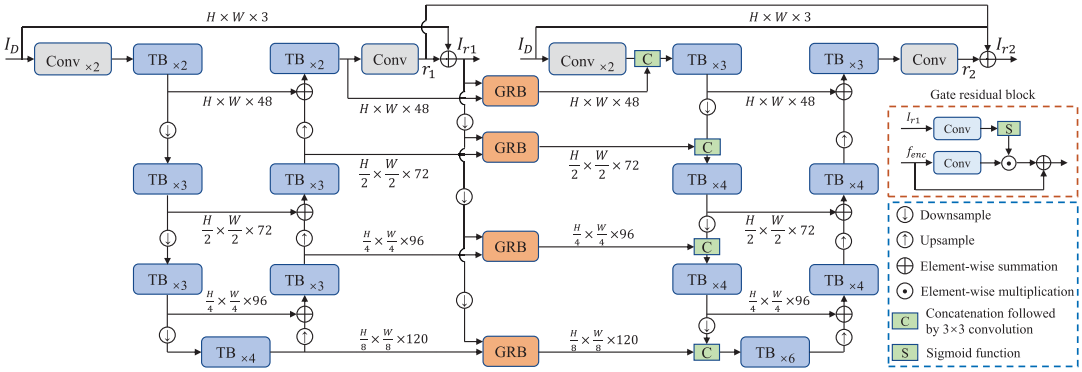


Fig. 3. Network architecture of the proposed CUTNet, which employs a cascaded U-Transformer architecture composed of Transformer blocks (TB) to overcome the limitations of CNN. To cascade the two U-Transformers, the decoder features of the first module are concatenated to the same-scale encoder features of the latter module after going through gate residual blocks (GRB) whose architecture is shown in the right inset.

Although these reference-based image SR and inpainting methods are effective, they cannot be directly applied to the reference-based MDIR task, because quality degradations can significantly prevent accurate content/texture matching between the reference and distorted images. In the following section, we describe our proposed multi-stage progressive restoration model which relies on a quality-degradation-restoration method to encourage similar amounts of distortion artifacts in the reference and distorted images, and thus leads to a more effective texture matching and transferring process, ultimately resulting in restored images of improved visual quality.

III. Algorithm

In this section, we describe the details of building and training the proposed Ref-IRT model. As shown in Figure 2, Ref-IRT progressively restores an image via three main stages: (1) preliminary restoration via a cascaded U-Transformer network; (2) primary restoration based on texture matching and transfer; and (3) final restoration for more elegant edge/texture reconstruction. We describe detailed network architectures of each stage in the following subsections.

A. Preliminary Restoration

The first stage of Ref-IRT performs the preliminary restoration on the distorted image such that similar edges/textures can be possibly and more accurately found in the reference image. Theoretically, any existing IR network can be retrained and used in this stage. However, we do notice that a more effective IR model will definitely benefit the restoration process in the subsequent stages. Motivated by [33] that stacks a number of Transformer blocks (TBs) into a U-Net shape to achieve the state-of-the-art IR performance, we propose a cascaded U-Transformer architecture that progressively approximates the residual image in an easy-to-hard and coarse-to-fine manner. Since different numbers of TBs were assigned to the two U-Transformer modules due to the different levels of residual-prediction accuracy, the network parameters of our model can be significantly reduced as compared with [33] while still achieving better/competitive performance.

The network architecture of CUTNet is illustrated in Figure 3, in which "GRB" denotes the gate residual block whose architecture is shown on the right side of the figure. For each U-Transformer module, two convolution layers with 3×3-pixel filter size and one-pixel zero padding are first applied to obtain the low-level feature embeddings. Then, these shallow features are processed by a four-level encoder-decoder, in which each level contains multiple Transformer blocks, all of which share the same architecture as that in [33]. Given an input of $H \times W \times 3$ pixels, the number of TBs and the dimension of the output features in each level are illustrated in Figure 3. Instead of using pixel unshuffle/shuffle to perform feature downsampling/upsampling as adopted in [33], we use bilinear interpolation followed by a 1×1 convolution layer to more flexibly downsample/upsample the feature maps to the desired channels. Finally, in the decoder, the convolution layer (with the same filter size and number of padding pixels) maps the $H \times W \times 48$-pixel feature maps to the residual image $r_i \in \mathbb{R}^{H \times W \times 3}$ $(i = 1, 2)$.

To cascade the two U-Transformer modules, the decoder features of the first module are concatenated to the same-scale encoder features of the latter module after going through the GRB. As shown in Figure 3, the encoder feature f_{enc} is re-calibrated by the gate-weight generated from the restored image I_{r1} such that the less informative features are suppressed and the more useful information is propagated. Instead of simply passing the output of one module to the other, each U-Transformer in our model has its own input (i.e., the distorted image I_D) and output (i.e., I_{r1}/I_{r2}), and the network is trained to minimize the loss between the ground-truth and the outputs of both modules [see Eqs. (15), (16)]. In this way, the residual image r_2 output by the second U-Transformer module can be viewed as a fine error compared with the coarse error r_1 in the first module if adding r_1 to the final output. Such a progressive residual learning and feature propagation mechanism can be generalized to operate multiple times in the case that the network contains multiple U-Transformer modules, i.e., $I_r = I_D + r_1 + r_2 + \cdots + r_n$. However, as we have found, using two U-Transformer modules is sufficient to reach a satisfactory restoration performance without a significant increase in the number of network parameters. After this stage, the preliminarily-recovered image will be continuously processed in the second stage to further improve its quality, which is described in the next subsection.

B. Primary Restoration

The second primary restoration stage aims to improve the restoration performance by transferring similar edges/textures from a reference image to the distorted image to help recover

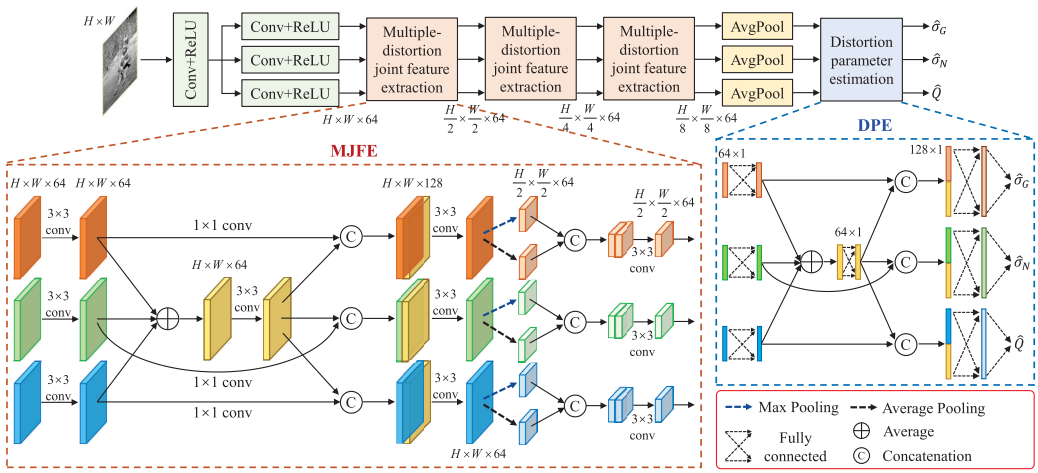


Fig. 4. Network architecture of the proposed MPENet, which estimates the three distortion parameters from the distorted image's luminance channel. MPENet employs the cascaded MJFE architecture, and each MJFE block uses the convolution, fusion, and concatenation operations to capture the characteristic features of the distortions and their intensities. The distortion parameters are then estimated by feeding the output features of the last MJFE block to the average pooling layer followed by the DPE block. Note that we use $\hat{\sigma}_G$, $\hat{\sigma}_N$, $\hat{Q}$ in the figure to denote the distortion parameters estimated by MPENet, and $\tilde{\sigma}_G$, $\tilde{\sigma}_N$, $\tilde{Q}$ in Eqs. (1)-(3) to indicate the ground truth.

the lost information. The key to this stage is to perform an accurate content/texture matching between the reference and the distorted images. Since a preliminarily-recovered image is not as perfect as the reference, and the distortion artifacts can significantly hinder the matching process, we propose a quality-degradation-restoration method to allow the reference image to display a similar level of distortion as the distorted image.

Specifically, a CNN model is first employed to predict the distortion parameters of the original multi-degraded image, based on which the same amounts of the multiple distortions are added to the reference image. Then, the quality-degraded reference image is processed by the same CUTNet model in Section III-A, and matching is conducted on the two preliminarily-recovered images: one preliminarily recovered image corresponds to the distorted image and the other preliminarily recovered image corresponds to the quality-degraded reference image. Finally, the fine edges/textures from the original reference image are transferred to the preliminarily-recovered distorted image to improve its perceived quality. The details of these steps are as follows:

1) Distortion Parameter Estimation: As in [44], the synthesized multi-degraded images are assumed to contain three distortion types: Gaussian blur, white noise, and JPEG compression. Specifically, given a pristine image, the blur distortion was first added by applying the image with a Gaussian filter whose standard deviation σ_G was randomly selected from the range $[0, 10]$. Then, the noise distortion was generated by adding each location on the R, G, B planes of the image by a value randomly sampled from zero-mean normal distributions whose variances σ_N^2 fall into the range $\left[\left(\frac{0}{255}\right)^2, \left(\frac{55}{255}\right)^2\right]$. Finally, the JPEG compression distortion was introduced by quantizing the discrete cosine transform (DCT) coefficients based on a quantization table whose values are determined by a quality parameter Q (i.e., the scaling level of compression) which ranges from 10 to 90.

To summarize, the parameters used to generate the three distortion types are (1) the standard deviation $\sigma_G \in [0, 10]$ for generating the Gaussian blur, (2) the variance $\sigma_N \in [0, \frac{55}{255}]$ for generating the white noise, and (3) the compression quality factor $Q \in [10, 90]$ for the JPEG compression. Instead of predicting the raw distortion parameters, we predict normalized values such that the network can be trained more consistently. Specifically, the target distortion parameter values used to train MPENet are given by

$$\tilde{\sigma}_G = \sigma_G/10 \tag{1}$$

$$\tilde{\sigma}_N = 255 \times \sigma_N/55 \tag{2}$$

$$\tilde{Q} = \sqrt{Q}/10 \tag{3}$$

where $\tilde{\sigma}_G$, $\tilde{\sigma}_N$, and $\tilde{Q}$ denote, respectively, the normalized Gaussian blur, white noise, and JPEG-compression distortion parameters.

The architecture of the proposed MPENet is illustrated in Figure 4. Specifically, the network takes as input a luminance (Y-channel) image and outputs three distortion parameters by using respectively three branches, each of which contains several convolution layers (Conv) and fully-connected (FC) layers. To model the joint effect of multiple distortions on predicting the distortion parameter of an individual distortion type, the feature maps of different branches are fused (averaged) and concatenated back to the corresponding branch after being processed by a convolution layer for feature nonlinearity. For each branch, the concatenated features are first convolved by filters of 3×3-pixel size to contain the half number of channels, and then downsampled by both the max and average pooling operations to extract the peak/average feature values over each non-overlapping 2×2-pixel region. These downsampled features are again concatenated and convolved to the same number of channels as the previous layer. In this way, after going through a multiple-distortion joint feature extraction (MJFE) block, the feature dimensions are changed from $[H, W, C]$ to $[\frac{H}{2}, \frac{W}{2}, C]$, where H, W, and C denote, respectively, the height, width, and channel number of the feature.

To enable the scalar output representing the different distortion parameter values, the output feature maps in each branch of the third MJFE block are collapsed into a single vector through average pooling. The three vectors are then passed through the distortion parameter estimation (DPE)

block which contains a number of the FC layers built in a similar fusion-and-concatenation manner to produce the final distortion parameter estimate. The only difference is that the convolution operation applied on feature maps is replaced by a linear projection applied on feature vectors. Given an image patch of $H \times W$ pixels, the dimension of each layer output is illustrated in Figure 4.

Since our goal is to predict distortion parameters of the entire image, MPENet is applied to each 128×128-pixel patch with 64-pixel overlap in the testing stage, and an average pooling operation is applied to collapse all parameter values into a scalar. Let $\hat{\xi}_m^p$ $(m = 1, 2, 3)$ denote the three distortion parameter values predicted by MPENet for each image patch p. Then, the distortion parameters predicted for the entire image are given by

$$\hat{\xi}_m = \frac{1}{N_p} \sum_{p=1}^{N_p} \xi_m^p, \tag{4}$$

where N_p denotes the total number of all the selected patches. These estimated distortion parameters serve as a guidance for adding the same level of the distortions to the reference image by using the same method in [44]. Then, the quality-degraded reference image is processed by the same CUTNet model in Section III-A, giving rise to the quality-restored reference image which will be used for content/texture matching in the subsequent step.

2) Texture Matching and Transfer: A block diagram of the texture matching and transfer method is illustrated at the bottom side of Figure 5, which employs a similar idea as that being used in many reference-based image SR works (e.g., [40], [41]). As shown in the figure, given the preliminarily-recovered distorted image I_1, the quality-restored reference image I_{R1}, and the original reference image I_R, the first step is to perform the content/texture matching between I_1 and I_{R1}. To this end, the pre-trained VGG19 model [57] is applied to extract the multi-scale deep features from both images. Then, by calculating patch feature similarity, the dense patch-wise matching is performed which aims at finding the most similar patch in I_{R1} given a specific patch in I_1.

Let f_K and f_Q denote the deep features extracted by one convolution layer of VGG19 from I_{R1} and I_1, respectively. First, the relevance $r_{i,j}$ between the i-th postion in f_K and the j-th position in f_Q is computed by

$$r_{i,j} = \left\langle \frac{f_K^{(i)}}{\left\| f_K^{(i)} \right\|}, \frac{f_Q^{(j)}}{\left\| f_Q^{(j)} \right\|} \right\rangle, \tag{5}$$

where $f_K^{(i)}$ and $f_Q^{(j)}$ denote VGG features corresponding to a 3×3-pixel patch centered at the i-th position in f_K and the j-th position in f_Q; $\langle \cdot \rangle$ denotes the cosine similarity. Then, we compute a hard-attention map P in which the j-th element p_j is given by

$$p_j = \arg\max_i (r_{i,j}). \tag{6}$$

The value of p_j can be viewed as a hard index, which represents the most relevant patch in I_{R1} to the j-th patch in I_1. Meanwhile, we also compute the confidence map C in which the j-th element c_j is given by

$$c_j = \max_i (r_{i,j}). \tag{7}$$

The value of c_j represents the relevance/similarity between the selected patch in I_{R1} and the j-th patch in I_1. Accordingly, for each patch j in I_1, the most relevant patch p_j with a confidence value c_j can be found in I_{R1}.

Finally, the transferred patch features corresponding to the j-th patch in I_1 is given by

$$f_T^{(j)} = c_j \cdot f_V^{(p_j)}. \tag{8}$$

where f_V denotes VGG features extracted from I_R. In our implementation, the eighth convolution layer in VGG19 is employed to extract features from I_1 and I_{R1} to compute P and C; the second, fourth, and eighth convolution layers are employed to extract features from I_R to build the transferred features. As in [40], the VGG19 network parameters will also be updated during the training stage such that the more effective texture features can be extracted. Given $I_1 \in \mathbb{R}^{H \times W \times 3}$, then the dimensions of the three transferred features are $f_{T1} \in \mathbb{R}^{H \times W \times 64}$, $f_{T2} \in \mathbb{R}^{\frac{H}{2} \times \frac{W}{2} \times 128}$, and $f_{T3} \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times 256}$. These features are then fed into the texture reconstruction network to produce the texture-enhanced image, which will be described in the next subsection.

3) Texture Reconstruction: The third step of the primary restoration stage is to map the transferred features to the high-quality image. To this end, a three-scale U-Transformer network was employed whose architecture is illustrated in the middle part of Figure 5. As shown in the figure, the network also consists of an encoder and a decoder. The encoder serves to extract the multi-scale deep features from I_1, and the decoder serves to combine the transferred features with features of I_1 to reconstruct the high-quality image. Different from the TB-based U-Transformer module in Section III-A, here we incorporate the shifted-window mechanism [22] within transposed attention (TA) [33] to build a dual-window

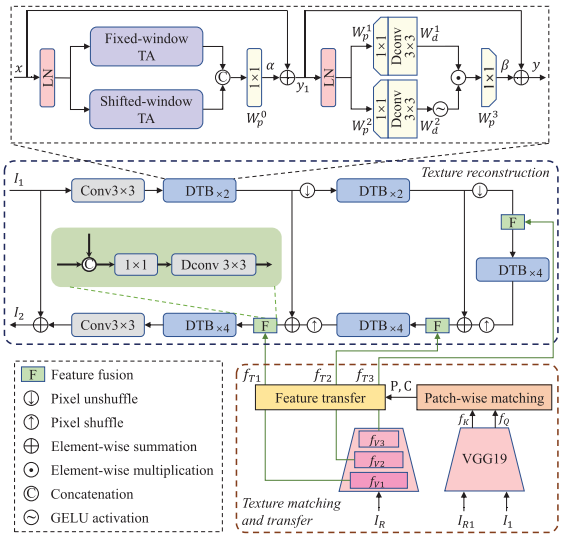


Fig. 5. Network architecture of the proposed texture matching, transfer, and reconstruction network. The dense patch-wise matching is first performed between I_{R1} and I_1 to determine the transferred features (f_{T1}, f_{T2}, f_{T3}) of I_R, which are then fed into the three-scale U-Transformer-based texture reconstruction network to produce the texture-enhanced image. Note that we use the dual-window Transformer block (DTB) whose architecture is illustrated in the topmost inset to facilitate the reconstruction.

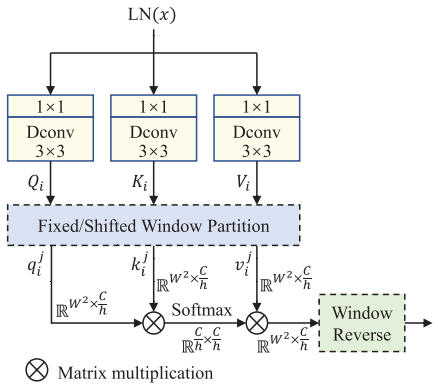


Fig. 6. Network architecture of the proposed fixed/shifted-window transposed attention mechanism. The layer-normalized (LN) feature input x is first processed by three independent projection Transformers, giving rise to the three features (Q_i, K_i, V_i) which are then split into windows to yield the window-based features (q_i^j, k_i^j, v_i^j). Then, the transposed attention for each window is computed, and the collection of the attention values are spatially aligned and concatenated to yield the final output.

Transformer block (DTB), which is more locally adaptive due to the different attention matrix computed for different local regions. The network architecture of DTB is illustrated in the top of Figure 5, and the details of the fixed/shifted-window transposed attention are provided in Figure 6.

Specifically, given an input $x \in \mathbb{R}^{H\times W\times C}$, three independent projection Transformers each of which consists of a 1×1-pixel convolution layer followed by a 3×3-pixel depth convolution (Dconv) layer are applied to the layer-normalized (LN) feature to obtain the three features $Q_i, K_i, V_i \in \mathbb{R}^{H\times W\times \frac{C}{h}}$ corresponding to the i-th head with head number h. These features are then split into $\frac{HW}{w^2}$ windows, where w denotes the window size. Let $q_i^j, k_i^j, v_i^j \in \mathbb{R}^{w^2\times \frac{C}{h}}$ denote the j-th window feature of Q_i, K_i, and V_i. Then, the transposed attention for each window is computed by

$$a_i^j = \text{Attn}\left[\text{LN}\left(x^j\right)\right] = v_i^j \cdot \text{softmax}\left(k_i^{j\text{T}} \cdot q_i^j/\tau\right), \quad (9)$$

where τ is a learnable scaling factor that controls the magnitude of the dot product of $k_i^{j\text{T}}$ and q_i^j before the Softmax function is applied. Finally, all a_i^j are spatially aligned by the window reverse operation to obtain the i-th head output $a_i \in \mathbb{R}^{H\times W\times \frac{C}{h}}$, and all a_i are then concatenated to obtain the output of the fixed/shifted-window transposed attention block $a \in \mathbb{R}^{H\times W\times C}$. Note that half of the heads use a fixed-window partition and the other half use a shifted-window partition [22]. Consequently, the final output of DTB is formulated as

$$y = y_1 + \beta W_p^3\left\{\phi\left[W_d^2 W_p^2\left(LN\left(y_1\right)\right)\right]\right\} \odot W_d^1 W_p^1\left(\text{LN}\left(y_1\right)\right), \quad (10)$$

where $y_1 = x + \alpha W_p^0 \text{Attn}\left[\text{LN}\left(x\right)\right]$; W_p^i $(i = 0, 1, 2, 3)$ denotes the weight of each of the four 1×1 pixel-wise convolution layers; W_d^j $(j = 1, 2)$ denotes the weight of each of the two 3×3 depth-wise convolution layers; $\odot$ denotes element-wise multiplication; ϕ denotes the GELU non-linearity; α and β are two learnable scaling factors.

To utilize the side information from the reference, the transferred features f_{Ti} $(i = 1, 2, 3)$ are respectively connected to the three-scale decoder features through the feature fusion block, which maps the concatenated features to the desired channels via two convolution layers: a 1×1 pixel-wise convolution layer and a 3×3 depth convolution layer. As shown in Figure 5, we set the number of DTB in each scale of the encoder to be two and in the decoder to be four in order to better emphasize the transferred features. Also, we set the feature channel number of DTB in the first, second, and third scales to be 48, 96, and 192, respectively. Accordingly, the head numbers are set to be 2, 4, and 8 for the corresponding three scales. Moreover, we use bias in layer norm and a window size of 12 for all DTBs. The final output of the texture reconstruction network is formulated as

$$I_2 = I_1 + E_1, \quad (11)$$

where E_1 denotes the residual image predicted by the TMTR model in the primary restoration stage.

C. Final Restoration

Although the quality of the preliminarily-recovered image I_1 has improved via the primary restoration stage, the texture matching in that stage is applied on I_1 and I_{R1}, both of which are of relatively lower quality due to the remaining distortions. Since distortion artifacts can have a negative impact on the accuracy of content/texture matching, and consequently on the effectiveness of the transferred features, we argue that the restored image can be further enhanced if texture matching is conducted on I_2 and the quality-improved reference image I_{R2}, both of which display more image details than I_1 and I_{R1}.

Accordingly, the same TMTR model used in Section III-B is applied to I_{R1} to obtain the quality-improved reference image I_{R2}. In this case, the input of VGG19 for patch-wise matching is I_{R1} only, which means that the self-attention is performed. Then, another TMTR model is applied on I_2, I_{R2}, and I_R, which transfers texture features from I_R to I_2 based on the hard-attention map and confidence map computed for I_2 and I_{R2}. Note that the second TMTR model shares the same network architecture as the first one in Section III-B, but with different weight/bias values. Accordingly, the output of the third restoration stage is formulated as

$$I_3 = I_2 + E_2, \quad (12)$$

where E_2 denotes the residual image predicted by the second TMTR model. Note that we could theoretically build more stages to continuously apply TMTR models on I_3, I_4, and I_5, etc. However, as we have found, the performance does not change significantly when more than three stages are applied. Thus, by balancing the performance improvement, computational cost, and the available hardware, only three stages are adopted in the final Ref-IRT model.

D. Loss Function

Based on the three stages of Ref-IRT, we have to sequentially train four network models: the CUTNet, MPENet, and the two TMTR models in the second and third stages, respectively. The loss function for training MPENet is L_1 loss which is given by

$$\mathcal{L} = \sum_{m=1}^{3}\left|\hat{\xi}_m^p - \xi_m^p\right|, \quad (13)$$

where ξ_m^p and $\hat{\xi}_m^p$ denote, respectively, the ground-truth and predicted distortion parameters for image patch p; $m = 1, 2, 3$ denotes the three distortion types.

The loss function for training the other networks includes both the mean square error (MSE) loss and the structural similarity (SSIM) [58] loss. The overall loss function can be interpreted as

$$\mathcal{L} = \lambda \cdot l_{MSE} + l_{SSIM}, \tag{14}$$

where $\lambda = 10^3$ is a parameter used to adjust the weights of the two losses.

Specifically, for training CUTNet, the MSE loss is defined as

$$l_{MSE} = \frac{1}{HW} \sum_{i=1}^{HW} \left\{ [I_{r1}(i) - I(i)]^2 + [I_{r2}(i) - I(i)]^2 \right\}, \tag{15}$$

where i is a spatial location index of the image; W and H are image width and height; I_{r1} and I_{r2} denote the output images of the two U-Transformer modules as shown in Figure 3; I denotes the ground-truth image. The SSIM loss is defined as

$$l_{SSIM} = 2 - \overline{\text{SSIM}\,(I, I_{r1})} - \overline{\text{SSIM}\,(I, I_{r2})}, \tag{16}$$

where $\overline{\text{SSIM}\,(I, I_{rk})}$ $(k = 1, 2)$ denotes the average value of $\text{SSIM}\,(I, I_{rk})$ which is computed by

$$\text{SSIM}(I, I_{rk}) = \frac{\left(2\mu_I \mu_{I_{rk}} + C_1\right)\left(2\sigma_I \sigma_{I_{rk}} + C_2\right)}{\left(\mu_I^2 + \mu_{I_{rk}}^2 + C_1\right)\left(\sigma_I^2 + \sigma_{I_{rk}}^2 + C_2\right)}. \tag{17}$$

Here, $\mu_{I/I_{rk}}$ and $\sigma_{I/I_{rk}}$ denote, respectively, the local mean and local standard deviation of I/I_{rk}; C_1 and C_2 are two constants which take the same values as in [58].

For training the two TMTR models, the MSE and SSIM loss functions are respectively computed by

$$l_{MSE} = \frac{1}{HW} \sum_{i=1}^{HW} [I_n(i) - I(i)]^2 \tag{18}$$

$$l_{SSIM} = 1 - \overline{\text{SSIM}\,(I, I_n)} \tag{19}$$

where I_n $(n = 2, 3)$ denotes the output images of the second and third stages of Ref-IRT as shown in Figure 2; the variables I, i, H, W have the same definitions as in Eq. (15).

IV. Experiments

In this section, we evaluate the IR performance of Ref-IRT on several multi-degraded image datasets. We also compare the performance of Ref-IRT with the state-of-the-art MDIR methods.

A. Implementation Details

1) Training Data: The training data consists of 800 images from the DIV2K dataset [59] and 11,871 image patch pairs from the CUFED5 training dataset [34]. Each pair in CUFED5 contains a pristine patch and a corresponding reference patch at the size of $160 \times 160 \times 3$ pixels. The DIV2K dataset was used to train CUTNet and MPENet, while the CUFED5 dataset was used to train the two TMTR models. Specifically, for each pristine image in DIV2K, we used the method described in Section III-B1 to generate the multi-degraded images at three different scales (one original scale and two down-sampled scales), giving rise to 2,400 pristine images as well as their associated multi-degraded versions. For each of the 2,400 pristine-distorted image pairs, the non-overlapping 128×128-pixel size patch pairs were created, which were used to train CUTNet. Note that when generating the distorted images, we also save the ground-truth distortion parameter values, which were used as the labels for the distorted patches to train MPENet. Consequently, we generated in total 189,735 patch pairs from DIV2K.

To train the two TMTR models, multiple distortions were first added to the 11,871 pristine patches in CUFED5. Since the patch size is small, every group of 9 patches were concatenated into a $480 \times 480 \times 3$-pixel size image, which was then simultaneously contaminated by the three distortion types at ten different levels, giving rise to 13,190 multi-degraded images. These $480 \times 480 \times 3$-pixel size images were then fed into the well-trained CUTNet in the first stage to produce the preliminarily-recovered images. They were also fed into the well-trained MPENet to produce the three estimated distortion parameters, which were used to generate the same levels of the three distortions to the corresponding $480 \times 480 \times 3$-pixel size reference images created in the same way as the pristine patch. These quality-degraded reference images were then processed by the same CUTNet model to produce the preliminarily-recovered reference images. All of the preliminarily-recovered distorted patches and reference patches, and the original reference patches were used to train the TMTR model. Consequently, a number of $118,710 \times 3$ patches were generated to train each TMTR model. Note that a similar approach was adopted to generate the training data for the TMTR model in the third stage. The difference is that the MPENet was no longer used because there is no need to generate the quality-degraded reference images. Also, it is the well-trained TMTR model in the second stage that was applied to the preliminarily-recovered distorted/reference images to generate the training patches, not the CUTNet applied to the distorted and quality-degraded reference images.

2) Testing Data: To evaluate the performance of Ref-IRT, we used as testing data the multi-degraded images generated from the pristine images in three datasets, among which CUFED5 [34] and WR_SR [42] are two existing datasets, and the third XRIR dataset is proposed in this paper, where images were collected from both the Internet and our own camera. Each dataset consists of the pristine images and the corresponding reference images that share similar contents/textures. Specifically, the CUFED5 testing dataset contains 126 pristine images, and each pristine image has five reference images corresponding to the five different similarity levels. The WR_SR and XRIR datasets contain, respectively, 80 and 200 pristine images, each of which has only one reference image. Sample pristine-reference image pairs in the proposed XRIR dataset are shown in Figure 7 (additional details are available in our online supplement at https://vinelab.jp/refmdir/). Again, we use the same approach in Section III-B1 to generate the multi-degraded version of each pristine image in the three datasets. Note that WR_SR and XRIR contain images of high resolutions, thus we downsampled the pristine and reference images in the two datasets to reasonable sizes[1] before adding distortions such that all compared methods can be tested on

[1]In this paper, given an image whose minimal height/width is larger than 512 pixels, the bicubic downsampling will be applied such that the minimal length of the height/width of the image are 512 pixels.

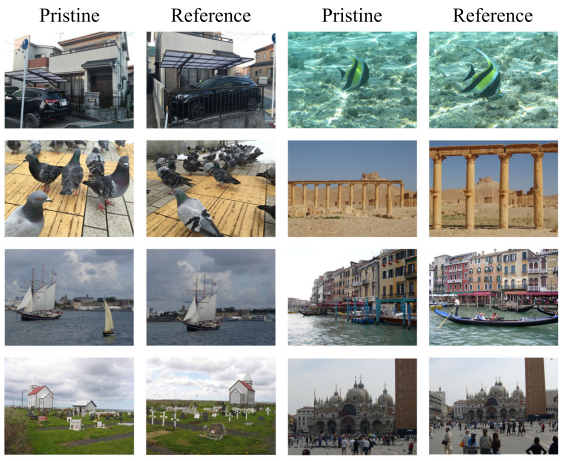


Fig. 7. Sample pristine-reference image pairs in the XRIR dataset.

the GPU without exhausting the memory. Consequently, the number of test multi-degraded images were 126, 80, 200 for CUFED5, WR_SR, and XRIR, respectively.

3) Parameter Settings and Network Training: We conducted all experiments by using the PyTorch framework on a remote server with an NVIDIA GeForce RTX 3090 GPU. Four separate models were sequentially trained on 128×128-pixel patches: CUTNet, MPENet, and the two TMTR networks. During the training, the network parameters were initialized with values sampled from a normal distribution $N(0, 0.02)$, and the leaky slopes were initialized to 0.1 for PReLU. We used Adam optimizer [60] with an initial learning rate of 2×10^{-4}, which was steadily decreased to 10^{-7} using the step decay. Specifically, the learning rate was scaled down by a factor of 0.9 after very epoch when training MPENet, and per 24,000 iterations when training CUTNet and TMTR networks. In total, the CUTNet was trained with a batch size of 12 for 15 epochs; MPENet was trained with a batch size of 64 for 80 epochs; and each TMTR network was trained with a batch size of 12 for 40 epochs. Consequently, the whole Ref-IRT model took about one week to train.

B. Algorithms and Performance Measures

We compared Ref-IRT with several state-of-the-art MDIR algorithms which include RL-Restore [44], OWAN [36], HOWAN [38], RMBN [35], and MEPS [37]. We also compared with DnCNN [61], DuRN [62], MIRNet [63], COLA-Net [64], SwinIR [31], and Restormer [33], all of which are state-of-the-art IR methods that can handle multiple degradations. In addition, we compared with Double-UNet [65], W-Net [66], and StackUNet [67]. Though initially designed for image segmentation, these models have similar network architectures as CUTNet. Finally, we compared with TTSR [40], RefVAE [49], MASA [41], and DATSR [43], all of which are reference-based image SR networks. For fair comparisons, we retrained most of these methods on our own data by using the same parameter settings except RL-Restore [44] which was pre-trained on DIV2K [59] using the same data-generation method. For the four reference-based SR methods, both the pre-released and the retrained models were used for testing.

Four criteria were used to measure the performance of each IR method: (1) peak signal-to-noise ratio (PSNR), (2) SSIM [58], (3) the learned perceptual image patch similarity (LPIPS) [68], and (4) the deep image structure and texture similarity (DISTS) [69], all of which have been widely used in previous IR studies. PSNR estimates image quality in terms of noise, while SSIM operates based on similarity measurements of three elements: luminance, contrast, and structure. Both LPIPS and DISTS operate by measuring the similarity between the reference and distorted images in the feature space. Based on the networks used to extract the deep features, the LPIPS index employed in this paper has two variants, denoted by LPIPS_A and LPIPS_V, respectively, which correspond to employing AlexNet [70] and VGG [57] for feature extraction. Note that both PSNR and SSIM make local comparisons on image pixels, while LPIPS and DISTS assess image quality at the patch level and thus allow more tolerance to texture resampling. Also note that SSIM was computed based on the luminance of the image, while the others were applied to the RGB color images. Higher PSNR/SSIM values and lower LPIPS/DISTS values indicate better image quality.

C. Overall Quantitative Results

1) Restoration Performance: Table I shows the four performance measures (averaged over all images) of Ref-IRT and other MDIR algorithms tested on the aforementioned dataset images simultaneously corrupted by the three distortion types: Gaussian blur, white noise, and JPEG compression. Except TTSR, RefVAE, MASA, and DATSR, all other MDIR algorithms are reference-free, meaning that the reference image is not required. Also included in Table I are the PSNR, SSIM, LPIPS, and DISTS values of the original multi-degraded images as well as the results of the first and second stages of Ref-IRT (denoted by "Ref-IRT-I" and "Ref-IRT-II", respectively) for comparison. In the test, both Ref-IRT-I and the reference-free MDIR methods were trained by using the same DIV2K dataset [59] images, while the others were trained by using the 11,871 pairs of pristine patches in the CUFED5 dataset [34]. Methods marked by "$*$" indicate that the pre-released models were used for testing. Also, the most similar reference image was used when testing the reference-based models on CUFED5. Results of the best-performing MDIR methods in Table I are bolded. We also conducted a small subjective study to quantify the visual improvement achieved by the different MDIR methods. More details are provided at https://vinelab.jp/refmdir/.

The network parameter numbers, floating point operations (FLOPs), and inference times of the different MDIR methods are shown in Table II. The FLOPs were computed based on 128×128 pixel size images for all methods except RL-Restore [44] which operates on image patches of 63×63 pixel size. The inference time was computed by averaging over ten images, each of which has two versions with different sizes (256×256 and 512×512 pixels). For the reference-based IR/SR models, the reference images were assumed to share the same size as the distorted image. Note that a tiny version of DATSR with fewer network parameters was employed to accelerate the training process, and thus the corresponding results in Table II are noteworthy. Also note that among the 22.814M parameters in Ref-IRT, 6.049M are for CUTNet, 2.297M for MPENet, and 7.234M for each of the two TMTR models.

TABLE I

PERFORMANCE OF REF-IRT VS. OTHER MDIR METHODS TESTED ON THE THREE DATASET IMAGES MEASURED IN TERMS OF FOUR OBJECTIVE QUALITY ASSESSMENT METRICS

Dataset	CUFED5					WR_SR					XRIR				
Method	PSNR	SSIM	LPIPS_A	LPIPS_V	DISTS	PSNR	SSIM	LPIPS_A	LPIPS_V	DISTS	PSNR	SSIM	LPIPS_A	LPIPS_V	DISTS
Distorted	19.202	0.381	0.768	0.625	0.379	20.885	0.397	0.771	0.600	0.352	21.279	0.417	0.754	0.621	0.363
RL-Restore* [44]	21.115	0.576	0.531	0.546	0.313	23.279	0.637	0.492	0.503	0.285	23.668	0.627	0.526	0.539	0.299
OWAN [36]	21.960	0.640	0.499	0.519	0.291	24.224	0.698	0.464	0.476	0.272	24.399	0.671	0.513	0.523	0.290
HOWAN [38]	20.775	0.634	0.541	0.558	0.311	21.182	0.691	0.543	0.558	0.312	20.489	0.663	0.614	0.629	0.339
RMBN [35]	21.431	0.583	0.527	0.547	0.298	23.513	0.637	0.505	0.529	0.277	23.739	0.616	0.546	0.559	0.293
MEPS [37]	21.855	0.632	0.526	0.524	0.295	24.103	0.692	0.478	0.470	0.266	24.349	0.666	0.525	0.516	0.284
DnCNN [61]	21.374	0.607	0.544	0.547	0.312	23.493	0.668	0.491	0.495	0.279	23.791	0.644	0.534	0.538	0.292
DuRN [62]	21.236	0.606	0.598	0.569	0.311	23.300	0.665	0.560	0.529	0.278	23.600	0.643	0.602	0.567	0.293
MIRNet [63]	22.224	0.650	0.480	0.492	0.271	24.475	0.706	0.442	0.447	0.251	24.726	0.679	0.495	0.496	0.268
COLA-Net [64]	21.977	0.638	0.514	0.514	0.286	24.155	0.696	0.471	0.466	0.259	24.439	0.670	0.518	0.516	0.276
SwinIR [31]	22.013	0.639	0.507	0.514	0.289	24.228	0.699	0.463	0.462	0.262	24.518	0.672	0.514	0.511	0.281
Restormer [33]	22.218	0.650	0.478	0.493	0.273	24.479	0.707	0.444	0.446	0.251	24.727	0.679	0.499	0.496	0.271
DoubleUNet [65]	21.515	0.616	0.555	0.542	0.311	23.722	0.680	0.498	0.484	0.285	23.897	0.651	0.556	0.538	0.303
W-Net [66]	21.876	0.636	0.493	0.511	0.283	24.124	0.697	0.449	0.455	0.255	24.380	0.669	0.503	0.506	0.274
StackUNet [67]	21.725	0.629	0.537	0.528	0.298	23.962	0.691	0.490	0.471	0.266	24.263	0.665	0.540	0.522	0.287
TTSR* [40]	16.590	0.271	0.707	0.646	0.364	17.773	0.275	0.704	0.645	0.355	18.559	0.310	0.656	0.632	0.343
TTSR [40]	21.748	0.635	0.491	0.508	0.279	23.717	0.683	0.478	0.476	0.265	24.170	0.661	0.518	0.517	0.279
RefVAE* [49]	16.021	0.278	0.550	0.630	0.310	16.903	0.273	0.598	0.629	0.323	17.576	0.295	0.558	0.624	0.294
RefVAE [49]	19.743	0.410	0.625	0.574	0.307	21.239	0.415	0.683	0.569	0.294	21.722	0.435	0.653	0.587	0.302
MASA* [41]	15.060	0.193	0.828	0.659	0.377	15.553	0.170	0.876	0.672	0.381	16.307	0.198	0.839	0.661	0.359
MASA [41]	20.391	0.441	0.599	0.552	0.305	21.697	0.434	0.662	0.561	0.299	22.282	0.458	0.624	0.573	0.301
DATSR* [43]	16.666	0.263	0.685	0.642	0.364	17.552	0.250	0.719	0.653	0.360	18.440	0.287	0.667	0.639	0.334
DATSR [43]	21.720	0.630	0.468	0.515	0.287	23.835	0.687	0.449	0.478	0.264	24.202	0.663	0.487	0.518	0.276
Ref-IRT-I	22.254	0.652	0.476	0.488	0.270	24.503	0.708	0.439	0.441	0.247	24.762	0.682	0.492	0.491	0.267
Ref-IRT-II	22.999	0.694	0.383	0.425	0.230	24.597	0.714	0.422	**0.435**	0.240	25.254	0.710	0.426	0.451	0.240
Ref-IRT	**23.118**	**0.702**	**0.363**	**0.407**	**0.217**	**24.607**	**0.715**	**0.421**	**0.435**	**0.239**	**25.356**	**0.717**	**0.412**	**0.439**	**0.232**

TABLE II

THE NETWORK PARAMETER NUMBERS, FLOPS, AND INFERENCE TIME (SECOND) OF REF-IRT VS. OTHER MDIR ALGORITHMS

Method	RL-Restore [44]	OWAN [36]	HOWAN [38]	RMBN [35]	MEPS [37]	DnCNN [61]	DuRN [62]	MIRNet [63]	COLA-Net [64]	SwinIR [31]
# of Params (M)	0.196	0.391	1.621	8.327	2.241	0.558	0.817	31.787	1.803	11.504
FLOPs ($\times 10^9$)	0.474	5.908	4.587	32.316	26.375	9.179	13.397	204.039	33.343	188.034
Time (256×256)	0.023	0.052	0.090	0.047	0.064	0.005	0.013	0.158	7.139	0.452
Time (512×512)	0.057	0.155	0.346	0.079	0.259	0.018	0.049	0.552	28.261	2.128
Method	Restormer [33]	DoubleUNet [65]	W-Net [66]	StackUNet [67]	TTSR [40]	RefVAE [49]	MASA [41]	DATSR [43]	Ref-IRT-I	Ref-IRT
# of Params (M)	26.112	29.289	6.131	12.921	6.730	37.430	4.028	6.017	6.049	22.814
FLOPs ($\times 10^9$)	35.248	13.491	5.668	31.661	24.710	16.740	22.502	24.065	14.209	74.290
Time (256×256)	0.090	0.012	0.005	0.048	0.024	0.010	0.042	0.166	0.083	0.503
Time (512×512)	0.372	0.031	0.021	0.099	0.109	0.036	0.136	0.588	0.308	1.397

As observed in both tables, Ref-IRT-I generally provides better PSNR, SSIM, LPIPS, and DISTS values as compared to all other MDIR methods on all dataset images considered, yet with a relatively small number of network parameters, FLOPs, and inference time compared to the two next-best performing methods MIRNet and Restormer. This fact demonstrates that the proposed cascaded U-Transformer network can more effectively restore multi-degraded images with only a lightweight computational complexity. By comparing with Ref-IRT-II and Ref-IRT, we observe an apparent performance improvement, demonstrating that by referring to a reference, the perceived quality of a preliminarily-recovered image can be further enhanced. In comparison, the relatively weak performances obtained by using both the pre-released and the retrained reference-based SR models suggest that textures in the reference and distorted images can be inaccurately matched if the two images exhibit different quality degradations. These results demonstrate the effectiveness of the proposed quality-degradation-restoration method in handling the texture matching and transferring tasks in the reference-based MDIR field. The increased network parameters, FLOPs, and inference time of Ref-IRT can be justified for applications that require higher image quality.

2) Performance With Different Reference Images: Since in Ref-IRT similar textures are transferred from reference images to distorted images, the content/texture similarity between the reference and the target distorted image can affect the overall performance. To evaluate the robustness/adaptiveness of our model to different reference images, we tested on CUFED5 by using five reference images each with a different similarity level as compared to the distorted image. The testing results corresponding to the second and third stages of Ref-IRT are presented in Table III, where level 1 indicates the highest similarity and level 5 indicates the least. To further investigate the impact of using unrelated reference images on the algorithm performance, we tested on CUFED5, WR_SR, and XRIR by using as the reference the pristine images from the LIVE [71], CSIQ [72], and CBSD68 [73] datasets, all of which contain completely different image contents. The testing results are shown in Table IV in which each entry represents the average performance value computed over all distorted images which were processed by using the same pristine image as the reference. The "Best" entry indicates the best performance that can be achieved by using one pristine image in the dataset, and the "Worst" entry indicates the opposite.

TABLE III

PERFORMANCE OF THE SECOND AND THIRD STAGES OF REF-IRT TESTED ON THE CUFED5 DATASET IMAGES BY USING FIVE DIFFERENT LEVELS OF THE REFERENCE IMAGES

Stage	Level	PSNR	SSIM	LPIPS_A	LPIPS_V	DISTS
Stage-II	1	22.999	0.694	0.383	0.425	0.230
	2	22.614	0.672	0.422	0.457	0.248
	3	22.532	0.668	0.431	0.464	0.252
	4	22.466	0.664	0.439	0.471	0.256
	5	22.384	0.659	0.452	0.479	0.262
Stage-III	1	23.118	0.702	0.363	0.407	0.217
	2	22.670	0.677	0.411	0.448	0.241
	3	22.577	0.671	0.422	0.457	0.247
	4	22.494	0.666	0.433	0.466	0.253
	5	22.396	0.660	0.449	0.477	0.261

TABLE IV

PERFORMANCE OF REF-IRT TESTED ON THE CUFED5 DATASET IMAGES BY USING REFERENCE IMAGES FROM DIFFERENT DATASETS

		CUFED5					WR_SR					XRIR				
Metrics		PSNR	SSIM	LPIPS_A	LPIPS_V	DISTS	PSNR	SSIM	LPIPS_A	LPIPS_V	DISTS	PSNR	SSIM	LPIPS_A	LPIPS_V	DISTS
Ref-IRT-I		22.254	0.652	0.476	0.488	0.270	24.503	0.708	0.439	0.441	0.247	24.762	0.682	0.492	0.491	0.267
LIVE	Best	22.276	0.655	0.463	0.488	0.266	24.481	0.709	0.434	0.444	0.246	24.725	0.682	0.484	0.492	0.265
	Worst	22.257	0.654	0.475	0.494	0.272	24.455	0.708	0.443	0.454	0.254	24.689	0.681	0.494	0.501	0.272
CSIQ	Best	22.275	0.655	0.465	0.488	0.267	24.482	0.709	0.435	0.444	0.245	24.733	0.682	0.484	0.491	0.264
	Worst	22.237	0.653	0.482	0.495	0.274	24.447	0.708	0.447	0.454	0.254	24.692	0.681	0.500	0.501	0.273
CBSD68	Best	22.276	0.655	0.464	0.489	0.267	24.579	0.712	0.433	0.444	0.246	24.736	0.682	0.484	0.492	0.264
	Worst	22.257	0.653	0.479	0.495	0.273	24.450	0.708	0.446	0.455	0.254	24.694	0.681	0.499	0.501	0.273

TABLE V

PERFORMANCE THE FIRST AND SECOND STAGES OF REF-IRT TESTED BY ELIMINATING/REPLACING WITH DIFFERENT NETWORK COMPONENTS

Dataset		CUFED5					WR_SR					XRIR				
Method	# of Params (M)	PSNR	SSIM	LPIPS_A	LPIPS_V	DISTS	PSNR	SSIM	LPIPS_A	LPIPS_V	DISTS	PSNR	SSIM	LPIPS_A	LPIPS_V	DISTS
w/o GRB+PRL	6.017	22.237	0.651	0.480	0.489	0.270	24.483	0.707	0.442	0.442	0.248	24.735	0.681	0.497	0.492	0.268
w/o GRB	6.017	22.236	0.651	0.480	0.489	0.271	24.476	0.707	0.444	0.442	0.249	24.741	0.680	0.497	0.492	0.269
w/o PRL	6.049	22.229	0.651	0.481	0.491	0.271	24.487	0.707	0.443	0.443	0.248	24.734	0.681	0.496	0.492	0.267
GRB+PRL	6.049	22.254	0.652	0.476	0.488	0.270	24.503	0.708	0.439	0.441	0.247	24.762	0.682	0.492	0.491	0.267
TB	7.231	22.940	0.690	0.385	0.428	0.231	24.587	0.713	0.422	0.436	0.240	25.204	0.708	0.429	0.454	0.241
w/o MPE	7.234	22.868	0.688	0.395	0.435	0.236	24.579	0.713	0.422	0.437	0.241	25.161	0.706	0.433	0.458	0.244
DTB	7.234	22.999	0.694	0.383	0.425	0.230	24.597	0.714	0.422	0.435	0.240	25.254	0.710	0.426	0.451	0.240

As can be observed from the two tables, the performance of Ref-IRT is indeed affected by using different reference images, which is as expected. This is also the reason why the performance improvement on the WR_SR dataset images is relatively small (as shown in Table I). Specifically, for reference images in CUFED5 which display at least some similarities with the distorted image, the performance of the second/third stage of Ref-IRT still improves upon the first stage (though minor for using the least similar reference images), suggesting that a reference image does benefit the MDIR task in this scenario. However, the performance varies or even decreases when reference images with different contents are used. This fact might suggest that transferring textures from an irrelevant reference image can sometimes be useless or even harmful, because the texture reconstruction network has to remove the erroneous textures/structures additionally introduced by the texture matching and transferring module (as shown in Figure 5). Yet, under the current situation that a similar reference image can possibly be obtained on the Internet by using the website search engines, our method still has the potential to benefit the IR field especially when images are highly degraded.

D. Ablation Study

We perform an ablation study to validate the importance of different network components in Ref-IRT. Specifically, for CUTNet, we examined the contributions of the proposed gate residual block (GRB) and progressive residual learning (PRL) method by training three variant CUTNet models: (1) without using GRB (denoted by "w/o GRB") in which case the decoder features of the first U-Transformer module are directly concatenated to the encoder features of the second U-Transformer module; (2) without using PRL (denoted by "w/o PRL") in which case the residual computed by the first U-Transformer module (r_1) will not be added to the output of the second U-Transformer module (i.e., $I_{r2} = I_D + r_2$); and (3) without using both GRB and PRL (denoted by "w/o GRB+PRL"). For TMTR, we verified the effectiveness of the proposed quality-degradation-restoration method and dual-window Transformer block by training two variant TMTR models in the second stage of Ref-IRT. One variant is without using MPENet (denoted by "w/o MPE") in which case the reference image is directly employed for content/texture matching. The other variant replaces DTB with TB that has been used in [33] (denoted by "TB"). The five variant models were trained by using the same training data and parameter settings. The network parameter numbers as well as the testing results on the three dataset images are shown in Table V. Note that for testing on CUFED5, the reference image with the highest similarity is employed for texture transfer. Also included in Table V are the results of the original algorithm for reference (denoted by "GRB+PRL" and "DTB", respectively).

As can be seen from Table V, the performance of CUTNet decreases when either GRB or PRL is removed, demonstrating that both network components are essential to the first-stage preliminary restoration task. The performance of the TMTR model also decreases when DTB is replaced by TB, demonstrating that the proposed dual-window Transformer block is superior to the conventional Transformer block in reconstructing high-quality image details from transferred features. Furthermore, we observe an obvious performance drop on CUFED5 and XRIR when the reference image is directly used for matching, demonstrating that our strategy of conducting content/texture matching between two preliminarily-recovered images is an effective approach, thanks to the accurate prediction of the distortion parameters by MPENet (please see our online supplement for details). Such a performance drop is not quite apparent when testing on WR_SR, which is as expected, because there are relatively larger gaps in contents/textures between the reference and target images in the dataset. These results demonstrate that all network components including GRB, PRL, MPENet, and DTB are essential elements in boosting the overall performance of Ref-IRT.

E. Representative Qualitative Results

In this section, we provide visual comparisons of different MDIR methods applied on sample multi-degraded images selected from the three testing datasets. We also provide visual comparisons of applying Ref-IRT by using different reference

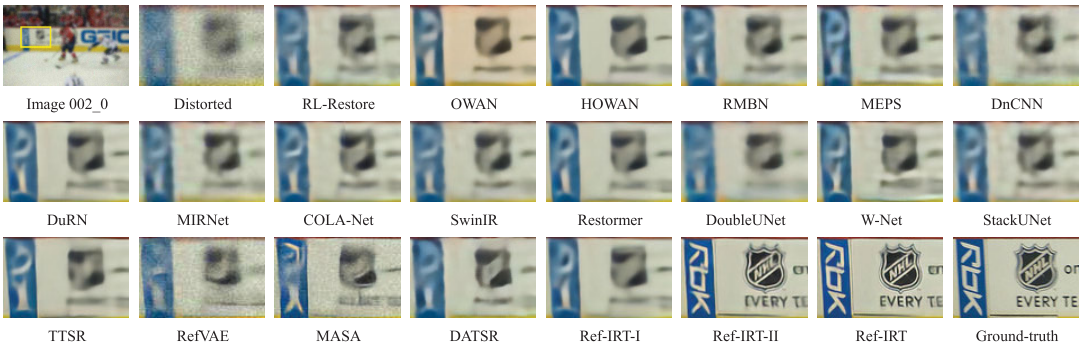


Fig. 8. Visual comparison of various MDIR methods applied on a sample multi-degraded image (*002_0.png*) in the CUFED5 dataset [34].

Fig. 9. Visual comparison of various MDIR methods applied on a sample multi-degraded image (*058.png*) in the WR_SR dataset [42].

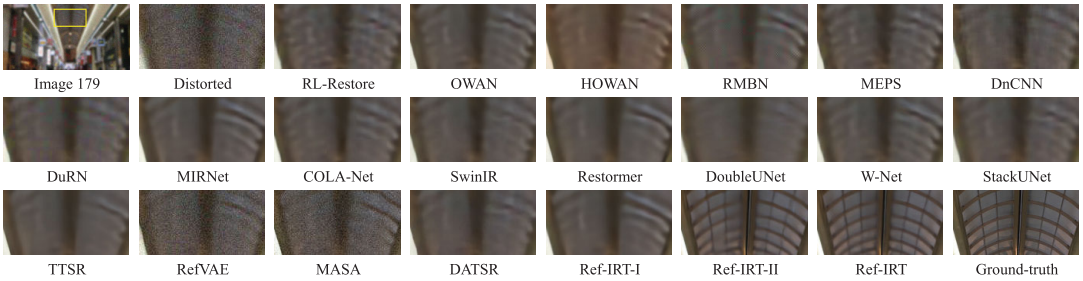


Fig. 10. Visual comparison of various MDIR methods applied on a sample multi-degraded image in the XRIR dataset.

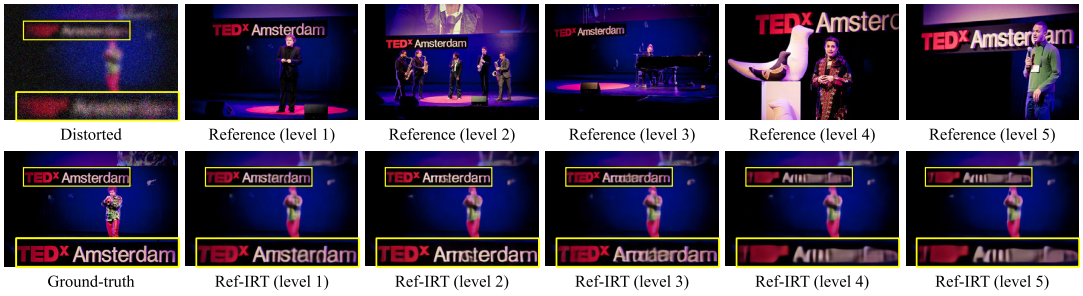


Fig. 11. Visual comparison of Ref-IRT applied on a sample multi-degraded image (*008_0.png*) in the CUFED5 dataset [34] by using different reference images. Note that the letters become less readable when the similarity between the reference and distorted images decreases.

images each with different similarity levels as compared with the target image. Again, level 1 indicates the highest similarity and level 5 indicates the least. The visual results produced by different MDIR methods are shown in Figures 8, 9, and 10 (the distorted image is cropped for better visualization), and the results of Ref-IRT using different references are shown in Figure 11. For all the figures, we include the distorted and ground-truth images for reference. For Figures 8, 9, and 10, we also include the results of the first and second stages of Ref-IRT for comparison.

As observed in the figures, our method can generate more favorable textures and sharper edges as compared to other methods. One the one hand, with the assistance of the texture features extracted from the reference, we observe that the perceived quality of the restored image can be significantly and progressively improved by each stage, especially when images are of high degradations with too much information changed/lost. In contrast, the restored images given by all state-of-the-art methods including Ref-IRT-I will inevitably give rise to the blur/ringing artifacts, which is attributed

TABLE VI

PERFORMANCE OF APPLYING REF-IRT$^+$ ON MULTI-DEGRADED IMAGES WITH DIFFERENT DISTORTION TYPES AND INTENSITIES

			Mild					Moderate					Severe				
Dataset	Dst. Type	Method	PSNR	SSIM	LPIPS_A	LPIPS_V	DISTS	PSNR	SSIM	LPIPS_A	LPIPS_V	DISTS	PSNR	SSIM	LPIPS_A	LPIPS_V	DISTS
CUFED5	B+J	Distorted	25.756	0.825	0.268	0.253	0.172	22.510	0.687	0.475	0.410	0.263	21.276	0.618	0.569	0.502	0.311
		Ref-IRT-I$^+$	28.284	0.889	0.148	0.213	0.116	25.160	0.791	0.293	0.333	0.186	23.327	0.713	0.413	0.422	0.244
		Ref-IRT$^+$	**28.893**	**0.905**	**0.105**	**0.193**	**0.097**	**26.249**	**0.832**	**0.199**	**0.277**	**0.141**	**24.638**	**0.775**	**0.273**	**0.337**	**0.176**
	N+J	Distorted	30.019	0.898	0.079	0.181	0.098	26.188	0.766	0.210	0.314	0.170	23.941	0.650	0.335	0.399	0.214
		Ref-IRT-I$^+$	**31.596**	**0.944**	**0.050**	**0.138**	**0.062**	29.137	0.904	0.087	0.204	0.090	27.843	0.875	0.111	0.245	0.108
		Ref-IRT$^+$	31.356	**0.944**	0.054	0.142	0.065	**29.370**	**0.913**	**0.080**	**0.190**	**0.083**	**28.218**	**0.889**	**0.097**	**0.221**	**0.094**
	B+N+J	Distorted	25.949	0.804	0.263	0.282	0.181	22.267	0.636	0.504	0.458	0.281	20.449	0.492	0.656	0.566	0.343
		Ref-IRT-I$^+$	28.270	0.882	0.145	0.214	0.116	24.841	0.776	0.305	0.351	0.194	22.958	0.693	0.434	0.445	0.256
		Ref-IRT$^+$	**28.929**	**0.901**	**0.104**	**0.193**	**0.096**	**25.872**	**0.817**	**0.213**	**0.293**	**0.148**	**24.061**	**0.748**	**0.303**	**0.366**	**0.191**
WR_SR	B+J	Distorted	30.044	0.886	0.192	0.182	0.120	24.932	0.737	0.403	0.355	0.222	23.533	0.674	0.493	0.456	0.272
		Ref-IRT-I$^+$	**31.729**	**0.927**	0.125	**0.177**	**0.095**	27.359	0.820	0.290	**0.310**	0.170	25.297	0.747	0.400	0.393	0.226
		Ref-IRT$^+$	31.196	0.925	**0.121**	0.194	0.101	**27.372**	**0.825**	**0.267**	0.311	**0.163**	**25.630**	**0.761**	**0.357**	**0.382**	**0.207**
	N+J	Distorted	32.604	0.901	0.146	0.205	0.094	27.526	0.739	0.326	0.341	0.163	24.726	0.608	0.463	0.429	0.204
		Ref-IRT-I$^+$	**34.482**	**0.958**	**0.057**	**0.131**	**0.057**	**31.336**	**0.919**	**0.091**	**0.192**	**0.081**	**29.730**	0.888	**0.118**	**0.235**	**0.096**
		Ref-IRT$^+$	33.577	0.955	0.068	0.149	0.069	30.925	0.918	0.096	0.204	0.087	29.481	**0.889**	0.119	0.242	0.099
	B+N+J	Distorted	28.745	0.835	0.260	0.264	0.151	24.374	0.658	0.489	0.432	0.247	22.184	0.494	0.653	0.546	0.319
		Ref-IRT-I$^+$	**30.945**	0.908	0.142	**0.195**	**0.101**	26.824	0.801	0.307	**0.331**	0.179	24.638	0.711	0.441	0.424	0.244
		Ref-IRT$^+$	30.693	**0.909**	**0.137**	0.209	0.105	**26.873**	**0.806**	**0.286**	0.332	**0.172**	**24.901**	**0.725**	**0.399**	**0.414**	**0.227**
XRIR	B+J	Distorted	29.287	0.864	0.238	0.246	0.154	25.326	0.722	0.451	0.418	0.247	23.946	0.655	0.537	0.512	0.291
		Ref-IRT-I$^+$	**31.403**	0.914	0.148	0.214	0.108	27.706	0.808	0.323	0.353	0.188	25.789	0.731	0.437	0.440	0.245
		Ref-IRT$^+$	31.368	**0.921**	**0.122**	**0.208**	**0.101**	**28.278**	**0.834**	**0.254**	**0.316**	**0.159**	**26.608**	**0.773**	**0.336**	**0.382**	**0.199**
	N+J	Distorted	31.923	0.898	0.124	0.230	0.115	27.324	0.742	0.298	0.376	0.187	24.608	0.607	0.442	0.459	0.230
		Ref-IRT-I$^+$	**33.873**	**0.952**	**0.059**	**0.149**	**0.062**	31.008	0.909	0.106	0.219	0.091	29.510	0.873	0.134	0.263	0.108
		Ref-IRT$^+$	33.480	**0.952**	0.063	0.159	0.068	**31.047**	**0.915**	**0.100**	**0.217**	**0.090**	**29.681**	**0.883**	**0.124**	**0.251**	**0.101**
	B+N+J	Distorted	28.286	0.816	0.291	0.324	0.182	24.485	0.626	0.535	0.501	0.276	22.680	0.514	0.652	0.582	0.327
		Ref-IRT-I$^+$	30.651	0.895	0.167	0.232	0.115	27.024	0.781	0.348	0.378	0.201	25.135	0.700	0.475	0.470	0.262
		Ref-IRT$^+$	**30.807**	**0.905**	**0.138**	**0.225**	**0.106**	**27.620**	**0.810**	**0.279**	**0.339**	**0.170**	**25.906**	**0.741**	**0.375**	**0.413**	**0.218**

to the irreversible nature of the image degradation process. In particular, even the retrained reference-based SR methods cannot produce the visually pleasant results, which is attributed to the inaccurate spatial alignment between the reference and distorted images caused by the quality degradation. One the other hand, by comparing results of Ref-IRT generated from different reference images, we observe different sharpnesses/readabilities of the letters, indicating that the content/texture similarity between the reference and distorted images do influence the algorithm performance. Although we are able to show only a limited number of demonstrative images, overall, the proposed method shows either superior or highly competitive restoration performance as compared to existing MDIR methods.

F. Model Generalizability

Despite the effectiveness of Ref-IRT in handling multi-degraded images, the distortion types we have so far discussed contain only simple distortions sequentially added to an image in a fixed order. Yet, in reality, real-world distortion can be more complex and also the quality degradation process can be more random. Thus, in this section, we additionally trained our method on images corrupted by using a more practical degradation model [46]. The generalizability of our approach was then investigated by analyzing its performance on different distortion types and intensities, and on real-world images as well.

1) Test on Different Distortion Scenarios: We followed [46] to synthesize practical image degradations. Specifically, the blur distortion was generated by filtering the images with two different Gaussian blur kernels: (1) the isotropic Gaussian kernel and (2) the anisotropic Gaussian kernel, and the kernel size was uniformly sampled from $\{7\times 7, 9\times 9, \cdots, 21\times 21\}$. The noise distortion was synthesized by using a three-dimensional zero-mean Gaussian noise model with covariance matrix Σ, which actually contains the general case and two special cases: (1) the channel-independent additive white Gaussian noise (AWGN) model corresponding to $\Sigma = \sigma^2\mathrm{I}$ where I is the identity matrix, and (2) the gray-scale AWGN model corresponding to $\Sigma = \sigma^2 1$ where 1 denotes a 3×3 matrix with all elements equal to one. The JPEG compression distortion was introduced by using the same approach in Section III-B1, but with a different quality parameter range, i.e., $Q \in [30, 95]$. As we are not dealing with image super-resolution, blur caused by downsampling/upsampling was not considered. Also, the JPEG compression was always used as the final degradation step because it occurs when images are finally saved in JPEG format. Thus, by incorporating a random shuffle strategy, the three distortion types considered were (1) blur + JPEG (B+J), (2) noise + JPEG (N+J), and (3) blur + noise + JPEG (B+N+J). Note that for blur distortion, either the isotropic Gaussian kernel or the anisotropic Gaussian kernel or both were used to generate the test images, while for noise distortion, the probabilities of applying the three different cases were set to 0.2, 0.4, and 0.4, respectively. Also note that for the "B+N+J" case, the blur and noise distortion can be shuffled in two different orders, i.e., B+N+J and N+B+J. We refer interested readers to [46] for more details.

To enable our approach to work on this new degradation model, the DPE block in MPENet has to be modified since more distortion parameters are required which include (1) the 2×2 covariance matrix Σ_G of the multivariate normal distribution for generating the isotropic/anisotropic Gaussian kernels; (2) the 3×3 covariance matrix Σ_N of the Gaussian noise model; and (3) the quality parameter Q for the JPEG compression. Since Σ_G and Σ_N are symmetric matrices, the number of parameters corresponding to the two distortions are three and six, respectively. Also, the sequential order of the two distortions being added to an image has to be predicted, because the blur distortion can help reduce the perceived noise strength. Accordingly, the DPE block was modified as follows: (1) the numbers of output nodes for the three branches were set to three, six, and one, respectively; and (2) some more FC layers were cascaded as a classifier which was fed by the same input vectors as the DPE block to predict the sequential order. We refer to this slightly modified version of our approach as Ref-IRT$^+$. More details about Ref-IRT$^+$ are provided in our online supplement at https://vinelab.jp/refmdir/.

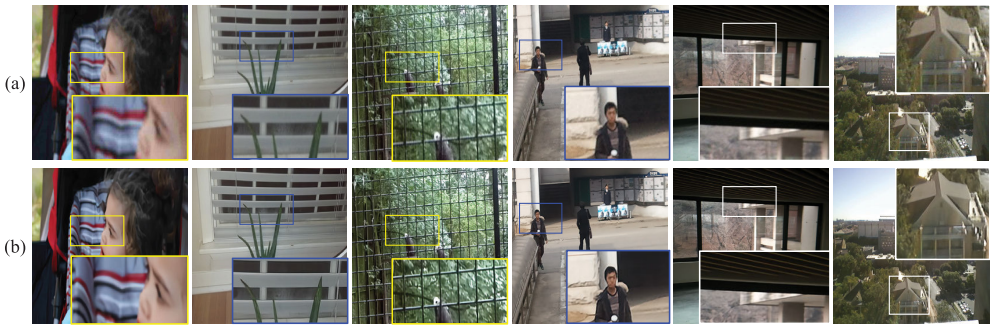


Fig. 12. Visual results of applying Ref-IRT$^+$ on sample images from the LIVE Challenge dataset [74]. Row (a) indicates the original inputs, and row (b) indicates the restored/enhanced images.

By changing the ten distortion parameter values, for each of the three distortion types, we synthesized three different distortion intensities (mild, moderate, and severe), and thus nine distortion scenarios were generated for each of the pristine images in CUFED5, WR_SR, and XRIR. Test results of Ref-IRT$^+$ on these distorted images are shown in Table VI, in which each entry represents the average performance measure computed for all the images in the dataset. Also included in Table VI are the PSNR/SSIM/LPIPS/DISTS values of the distorted images and the restored images obtained by applying the first stage of Ref-IRT$^+$ only (denoted by Ref-IRT-I$^+$). The best results for each distortion type at each distortion intensity are bolded.

From Table VI, we can draw two conclusions. First, by comparing the different distortion types, we observe that the reference image is less likely to help when images are corrupted by noise and JPEG compression. We suspect that this finding might be attributed to the information-additive properties of the two distortions, since transferred textures mainly help to supplement the middle/high-frequency information of an image lost in blurring. Second, by comparing the different distortion intensities, we observe that the reference image is not always necessary when images are mildly distorted. This is as expected, because a Transformer network might be good enough for recovering the mildly-distorted image contents, in which case the transferred textures might be less important, and can even harm the performance when textures from the two images differ significantly. Despite these potential limitations, we do observe that in most cases, the proposed reference-based strategy can improve upon the reference-free IR method especially when images are highly degraded.

2) Test on Real-World Distortion: We also tested Ref-IRT$^+$ on real-world distortions. To this end, images of lower qualities from the LIVE Challenge dataset [74] were used for testing, and the reference images were randomly selected from the 127 pristine images in the LIVE [71], CSIQ [72], and CBSD68 [73] datasets. As the ground-truth images are unavailable, we only show visual results of the sample restored/enhanced images in Figure 12 without providing their quality measures (more results are available at https://vinelab.jp/refmdir/). As can be observed, our method works quite well in reducing the noise, blur, and compression artifacts in these images, generally resulting in more visually pleasant results with sharp edges and clear textures. However, as it remains an open research challenge to automatically determine the most useful reference image(s) whose contents/textures are most relevant to the original images, the important role of the reference images in this test is not quite impressive; addressing this challenge falls into the range of our future work.

V. Conclusion

In this paper, we presented a three-stage framework (Ref-IRT) for reducing distortion artifacts in multi-degraded images. Our method operates by first conducting a preliminary restoration on the distorted image followed by the texture matching and transfer from a reference image to further enhance the restoration performance. The preliminary restoration stage employs a cascaded U-Transformer network for progressive residual learning such that a decent IR performance can be achieved with relatively small number of network parameters. The primary restoration stage induces a similar distortion to the reference image by referring to the estimated distortion parameters, and then applies restoration on the quality-degraded image such that the content/texture features between the reference and target images can be more accurately matched. Moreover, based on the matching result, a DTB-based U-Transformer network is proposed for high-quality image reconstruction. In the final restoration stage, the image quality is further enhanced by reapplying the matching procedure on the primarily-restored reference/distorted images, thus achieving the state-of-the-art IR performance. Experimental results tested on three benchmark datasets demonstrate the effectiveness of the proposed method. Future work could involve developing a more powerful texture matching, transfer, and reconstruction network to further improve the restoration performance. Future work could also involve taking into account additional real-world distortions and developing more efficient algorithms (e.g., by using semantic matching or with the assistance of the large language model) to automatically search for appropriate reference images such that the practicability of the proposed reference-based MDIR method can be further improved.

References

[1] X.-L. Zhao, W. Wang, T.-Y. Zeng, T.-Z. Huang, and M. K. Ng, "Total variation structured total least squares method for image restoration," *SIAM J. Sci. Comput.*, vol. 35, no. 6, pp. B1304–B1320, Jan. 2013.

[2] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[3] J. Zhang, D. Zhao, and W. Gao, "Group-based sparse representation for image restoration," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3336–3351, Aug. 2014.

[4] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, Jan. 2010.

[5] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2862–2869.

[6] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2005, pp. 60–65.

[7] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1620–1630, Apr. 2013.

[8] W. Dong, G. Shi, and X. Li, "Image deblurring with low-rank approximation structured sparse representation," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, Dec. 2012, pp. 1–5.

[9] X. Liu, X. Wu, J. Zhou, and D. Zhao, "Data-driven soft decoding of compressed images in dual transform-pixel domain," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1649–1659, Apr. 2016.

[10] H. Wang, Y. Cen, Z. He, Z. He, R. Zhao, and F. Zhang, "Reweighted low-rank matrix analysis with structural smoothness for image denoising," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 1777–1792, Apr. 2018.

[11] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[13] G. Lin, Q. Wu, L. Qiu, and X. Huang, "Image super-resolution using a dilated convolutional neural network," *Neurocomputing*, vol. 275, pp. 1219–1230, Jan. 2018.

[14] J. Dai et al., "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 764–773.

[15] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4700–4708.

[16] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, vol. 9351, 2015, pp. 234–241.

[17] I. Goodfellow et al., "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[18] L. R. Medsker and L. Jain, "Recurrent neural networks," *Design Appl.*, vol. 5, pp. 64–67, May 2001.

[19] X. Zhang, R. Jiang, T. Wang, and J. Wang, "Recursive neural network for video deblurring," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 8, pp. 3025–3036, Aug. 2021.

[20] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.

[21] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–22.

[22] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10012–10022.

[23] S. Ayas and E. Tunc-Gormus, "SpectralSWIN: A spectral-Swin transformer network for hyperspectral image classification," *Int. J. Remote Sens.*, vol. 43, no. 11, pp. 4025–4044, Jun. 2022.

[24] X. Huang, M. Dong, J. Li, and X. Guo, "A 3-D-Swin transformer-based hierarchical contrastive learning method for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5411415.

[25] H. Gong et al., "Swin-transformer-enabled YOLOv5 with attention mechanism for small object detection on satellite images," *Remote Sens.*, vol. 14, no. 12, p. 2861, Jun. 2022.

[26] Z. Liu, Y. Tan, Q. He, and Y. Xiao, "SwinNet: Swin transformer drives edge-aware RGB-D and RGB-T salient object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 7, pp. 4486–4497, Jul. 2022.

[27] J. Ma, L. Tang, F. Fan, J. Huang, X. Mei, and Y. Ma, "SwinFusion: Cross-domain long-range learning for general image fusion via Swin transformer," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 7, pp. 1200–1217, Jul. 2022.

[28] Z. Wang, Y. Chen, W. Shao, H. Li, and L. Zhang, "SwinFuse: A residual Swin transformer fusion network for infrared and visible images," 2022, *arXiv:2204.11436*.

[29] A. Lin, B. Chen, J. Xu, Z. Zhang, G. Lu, and D. Zhang, "DS-TransUNet: Dual Swin transformer U-Net for medical image segmentation," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–15, 2022.

[30] A. Hatamizadeh, V. Nath, Y. Tang, D. Yang, H. R. Roth, and D. Xu, "Swin UNETR: Swin transformers for semantic segmentation of brain tumors in MRI images," in *Proc. Int. MICCAI Brainlesion Workshop*, 2022, pp. 272–284.

[31] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van G., and R. Timofte, "SwinIR: Image restoration using Swin transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV) Workshops*, Oct. 2021, pp. 1833–1844.

[32] J. Fu et al., "Dual attention network for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3146–3154.

[33] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, and M.-H. Yang, "Restormer: Efficient transformer for high-resolution image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5728–5739.

[34] Z. Zhang, Z. Wang, Z. Lin, and H. Qi, "Image super-resolution by neural texture transfer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7982–7991.

[35] X. Liu, M. Suganuma, X. Luo, and T. Okatani, "Restoring images with unknown degradation factors by recurrent use of a multi-branch network," 2019, *arXiv:1907.04508*.

[36] M. Suganuma, X. Liu, and T. Okatani, "Attention-based adaptive selection of operations for image restoration in the presence of unknown combined distortions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Apr. 2019, pp. 9039–9048.

[37] S. Kim, N. Ahn, and K.-A. Sohn, "Restoring spatially-heterogeneous distortions using mixture of experts network," in *Proc. Asian Conf. Comput. Vis.*, 2020, pp. 1–226.

[38] Z. Huang, C. Li, F. Duan, and Q. Zhao, "Multi-distorted image restoration with tensor 1×1 convolutional layer," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2021, pp. 1–8.

[39] W. Shin, N. Ahn, J.-H. Moon, and K.-A. Sohn, "Exploiting distortion information for multi-degraded image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 536–545.

[40] F. Yang, H. Yang, J. Fu, H. Lu, and B. Guo, "Learning texture transformer network for image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5791–5800.

[41] L. Lu, W. Li, X. Tao, J. Lu, and J. Jia, "MASA-SR: Matching acceleration and spatial adaptation for reference-based image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6368–6377.

[42] Y. Jiang, K. C. K. Chan, X. Wang, C. C. Loy, and Z. Liu, "Robust reference-based super-resolution via C2-matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2103–2112.

[43] J. Z. Cao et al., "Reference-based image super-resolution with deformable attention transformer," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 325–342.

[44] K. Yu, C. Dong, L. Lin, and C. C. Loy, "Crafting a toolchain for image restoration by deep reinforcement learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2443–2452.

[45] X. Li et al., "Learning disentangled feature representation for hybrid-distorted image restoration," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 313–329.

[46] K. Zhang, J. Liang, L. Van Gool, and R. Timofte, "Designing a practical degradation model for deep blind image super-resolution," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 4791–4800.

[47] H. Zheng, M. Ji, H. Wang, Y. Liu, and L. Fang, "CrossNet: An end-to-end reference-based super resolution network using cross-scale warping," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 88–104.

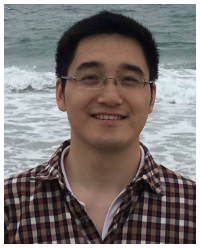
[48] A. Dosovitskiy et al., "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2758–2766.

[49] Z.-S. Liu, W.-C. Siu, and L.-W. Wang, "Variational autoencoder for reference based image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 516–525.

[50] Y. Xie, J. Xiao, M. Sun, C. Yao, and K. Huang, "Feature representation matters: End-to-end learning for reference-based image super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 230–245.

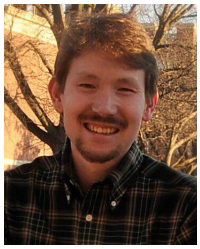
[51] L. Zhang, X. Li, D. He, F. Li, E. Ding, and Z. Zhang, "LMR: A large-scale multi-reference dataset for reference-based super-resolution," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 13118–13127.

[52] Y. Zhou, C. Barnes, E. Shechtman, and S. Amirghodsi, "TransFill: Reference-guided image inpainting by merging multiple color and spatial transformations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2266–2276.
[53] T. Liu, L. Liao, Z. Wang, and S. Satoh, "Reference-guided texture and structure inference for image inpainting," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2022, pp. 1996–2000.
[54] Y. Zhao, C. Barnes, Y. Zhou, E. Shechtman, S. Amirghodsi, and C. Fowlkes, "GeoFill: Reference-based image inpainting with better geometric understanding," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 1776–1786.
[55] D. Yoon, J. Kwak, Y. Li, D. Han, Y. Jin, and H. Ko, "Reference guided image inpainting using facial attributes," 2023, *arXiv:2301.08044*.
[56] R. Yasarla, H. R. V. Joze, and V. M. Patel, "Network architecture search for face enhancement," 2021, *arXiv:2105.06528*.
[57] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[58] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[59] E. Agustsson and R. Timofte, "NTIRE 2017 challenge on single image super-resolution: Dataset and study," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 126–135.
[60] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[61] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.
[62] X. Liu, M. Suganuma, Z. Sun, and T. Okatani, "Dual residual networks leveraging the potential of paired operations for image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7000–7009.
[63] S. W. Zamir et al., "Learning enriched features for real image restoration and enhancement," in *Proc. Eur. Conf. Comput. Vis.*, vol. 12370, 2020, pp. 492–511.
[64] C. Mou, J. Zhang, X. Fan, H. Liu, and R. Wang, "COLA-Net: Collaborative attention network for image restoration," *IEEE Trans. Multimedia*, vol. 24, pp. 1366–1377, 2022.
[65] D. Jha, M. A. Riegler, D. Johansen, P. Halvorsen, and H. D. Johansen, "DoubleU-Net: A deep convolutional neural network for medical image segmentation," in *Proc. IEEE 33rd Int. Symp. Comput.-Based Med. Syst. (CBMS)*, Jul. 2020, pp. 558–564.
[66] X. Xia and B. Kulis, "W-Net: A deep model for fully unsupervised image segmentation," 2017, *arXiv:1711.08506*.
[67] A. Sevastopolsky, S. Drapak, K. Kiselev, B. M. Snyder, J. D. Keenan, and A. Georgievskaya, "Stack-U-Net: Refinement network for improved optic disc and cup image segmentation," in *Medical Imaging 2019: Image Processing*. Bellingham, WA, USA: SPIE, 2019, pp. 576–584.
[68] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 586–595.
[69] K. Ding, K. Ma, S. Wang, and E. P. Simoncelli, "Image quality assessment: Unifying structure and texture similarity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2567–2581, May 2022.
[70] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.
[71] H. R. Sheikh, Z. Wang, A. C. Bovik, and L. K. Cormack. *Image and Video Quality Assessment Research at Live*. [Online]. Available: http://live.ece.utexas.edu/research/quality/
[72] D. M. Chandler, "Most apparent distortion: Full-reference image quality assessment and the role of strategy," *J. Electron. Imag.*, vol. 19, no. 1, Jan. 2010, Art. no. 011006.
[73] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th IEEE Int. Conf. Comput. Vis.*, Jul. 2001, pp. 416–423.
[74] D. Ghadiyaram and A. C. Bovik, "Massive online crowdsourced study of subjective and objective picture quality," *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 372–387, Jan. 2016.



Yi Zhang received the B.S. and M.S. degrees in electrical engineering from Northwestern Polytechnical University, Xi'an, China, in 2008 and 2011, respectively, and the Ph.D. degree in electrical engineering from Oklahoma State University, Stillwater, OK, USA, in 2015. From 2016 to 2018, he was a Postdoctoral Research Associate with the Department of Electrical and Electronic Engineering, Shizuoka University, Japan. He is currently an Associate Professor with the School of Information and Communications Engineering, Xi'an Jiaotong University, China. His research interests include 2D/3D image processing, machine learning, pattern recognition, and computer vision.

Qixue Yang received the B.S. degree in electronic information engineering from Chengdu University of Information Technology, Chengdu, China, in 2021, and the M.S. degree in communication engineering from Xi'an Jiaotong University, Xi'an, China, in 2024. His research interests include image super-resolution and denoising.



Damon M. Chandler (Senior Member, IEEE) received the B.S. degree in biomedical engineering from Johns Hopkins University, Baltimore, MD, USA, in 1998, and the M.Eng., M.S., and Ph.D. degrees in electrical engineering from Cornell University, Ithaca, NY, USA, in 2000, 2003, and 2005, respectively. From 2005 to 2006, he was a Postdoctoral Research Associate with the Department of Psychology, Cornell University. From 2006 to 2015, he was a Faculty Member with the School of Electrical and Computer Engineering, Oklahoma State University, USA. From 2016 to 2020, he was an Associate Professor with the Department of Electrical and Electronic Engineering, Shizuoka University, Japan. He is currently a Professor with the College of Information Science and Engineering, Ritsumeikan University, Japan. His research interests include image processing, data compression, computational vision, natural scene statistics, and visual perception.

Xuanqin Mou (Senior Member, IEEE) has been with the School of Electronic and Information Engineering, Institute of Image Processing and Pattern Recognition (IPPR), Xi'an Jiaotong University, Xi'an, China, since 1987, where he has been an Associate Professor since 1997 and a Professor since 2002. He is currently the Director of IPPR and the Director of the National Data Broadcasting Engineering and Technology Research Center. He has authored or co-authored over 200 peer-reviewed journals or conference papers. He was a member of the 12th Expert Evaluation Committee for the National Natural Science Foundation of China and the Executive Committee Member of China Society of Image and Graphics and Chinese Society for Stereology. He was also the Director of the Intelligent Imaging Society for Chinese Stereology. He was a recipient of the Yung Wing Award for Excellence in Education, the KC Wong Education Award, the Technology Academy Award for Invention by the Ministry of Education of China, and the Technology Academy Awards from the Government of Shaanxi Province, China.