Image Restoration by Estimating Frequency Distribution of Local Patches – Supplementary Material –

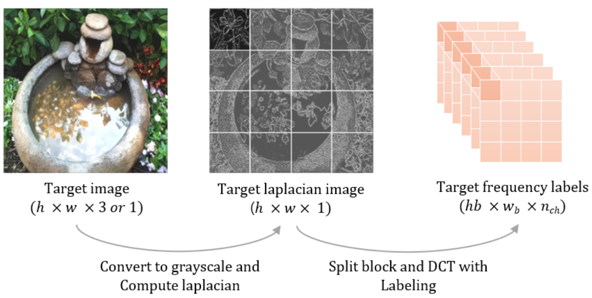

Figure 1: Data pre-processing for frequency labeling

1. Data Preprocessing

Local Blocks The proposed method changes the target image I^G from the spatial domain to the frequency domain to obtain the DCT coefficients q. However, when the image is analyzed in the frequency domain, since the location information of the data on the spatial axis is structurally lost, we divide I^G into local blocks.

DCT We use discrete cosine transform (DCT), which is widely used to convert images into frequency domain. Each block b_i is converted into n_{ch} channels of DCT coefficients $q^i = \{q^i_1, \cdots, q^i_{n_{ch}}\}$. The DCT coefficient q^i_{ch} expresses the frequency component in each frequency channel (band) and n_{ch} is inevitably $w_b \times h_b$.

Labeling To restore this DCT coefficient q^i_{ch} through classification, we need to label the continuous value q^i_{ch} to one of n_{cl} discrete values. For each frequency channel, we find a set of quantized DCT coefficients $m_{ch} = \{m_{ch,1}, ...m_{ch,n_{cl}}\}$ by dividing the support of the distribution of $\{q^i_{ch}\}^{n_b}_{i=1}$ in the training data set so that each bin has the same number of samples. Using m_{ch}, the target class label y^i_{ch} is obtained as follow:

$$y^i_{ch} = \underset{l\in\{1,...,n_{cl}\}}{\operatorname{argmin}} \left(\|m_{ch,l} - q^i_{ch}\|\right). \tag{1}$$

Laplacian Image In the above processes, we use the Laplacian image I^L of the original image I^G instead of using I^G to highlight the details of an image. In our method, the wider the range of the DCT coefficient, the larger becomes the quantization error. Using I^L that has a smaller DCT coefficient range than I^G, we can reduce the quantization error and also can focus on detailed texture.

2. The lower bound of BEF/MSE

For quantitative analysis, we have wanted to know the mean of BEF/MSE on the test set of LIVE1 and BSDS500, but we only had an information on the mean PSNR and the mean PSNR-B from the original papers. However, the lower bound of $\mathbb{E}$[BEF/MSE] can be derived as follows, which is reported in our paper. The PSNR ($P(y, \hat{y})$) and PSNR-B ($PB(y, \hat{y})$) of an image pair y and $\hat{y}$ are calculated as follows :

$$\begin{aligned} P(y,\hat{y}) &= 10\log_{10}\frac{255^2}{MSE(y,\hat{y})} \\ PB(y,\hat{y}) &= 10\log_{10}\frac{255^2}{MSE(y,\hat{y})+BEF(\hat{y})}. \end{aligned} \tag{2}$$

The mean PSNR and the mean PSNR-B of test set that other works reported are calculated as follows:

$$\begin{aligned} \mathbb{E}[P] &= \frac{1}{N}\sum_{i=1}^{N} P_i \\ \mathbb{E}[PB] &= \frac{1}{N}\sum_{i=1}^{N} PB_i, \end{aligned} \tag{3}$$

where N is the number of test samples. From the equation (2) the difference of mean PSNR and mean PSNR-B can be obtained

$$\mathbb{E}[P] - \mathbb{E}[PB] = \mathbb{E}[P - PB] = 10\,\mathbb{E}[\log_{10}(1 + \frac{BEF}{MSE})] \tag{4}$$

and by the Jensen's inequality, it becomes

$$\mathbb{E}[\log_{10}(1 + \frac{BEF}{MSE})] \le \log_{10}\mathbb{E}[1 + \frac{BEF}{MSE}]. \tag{5}$$

Combining (4) and (5), the lower bound of $\mathbb{E}[BEF/MSE]$ is obtained as

$$\mathbb{E}[\frac{BEF}{MSE}] \ge 10^{\frac{\mathbb{E}[P]-\mathbb{E}[PB]}{10}} - 1. \tag{6}$$

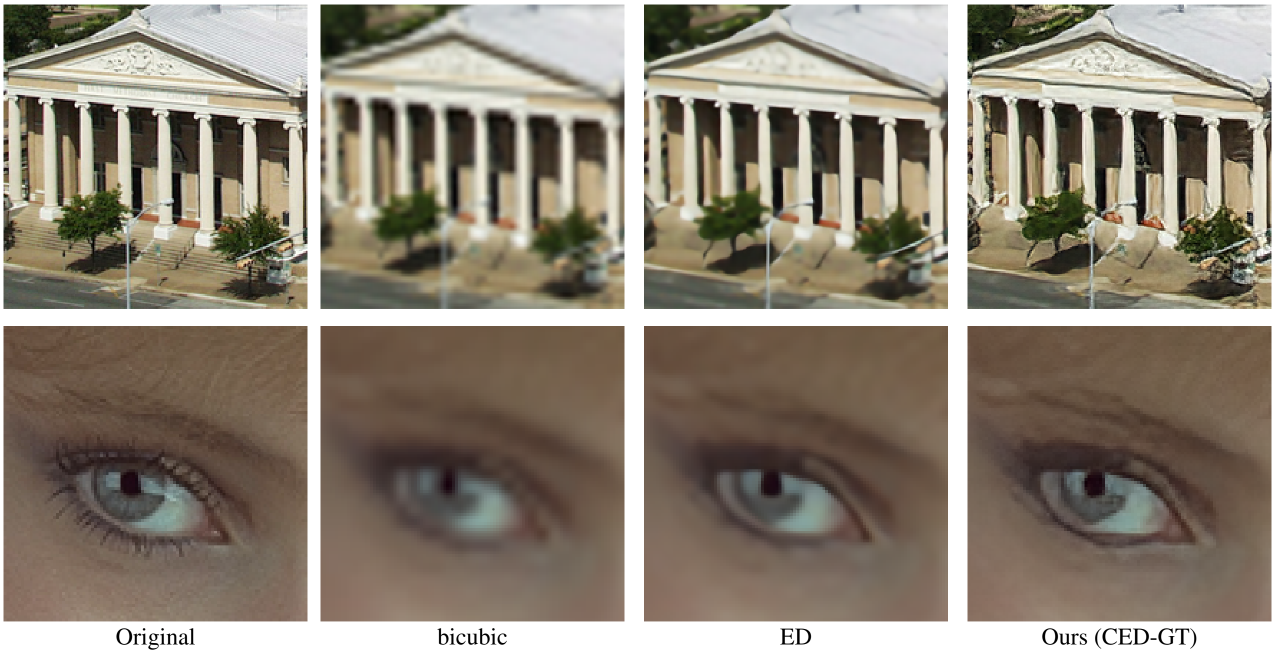

Figure 2: Comparison of super resolution results for LIVE1[2]. From left to right: Original, bicubic, ED, Ours (CED-GT).

3. Super Resolution

Our method can easily be applied to other image restoration tasks. One good example is super resolution, so we also experimented our method to the task of super resolution. For this task, we used the same datasets as compression artifact removal for both training and testing, but input and output dimensions of the networks are slightly different. We resized each training image to 256 $\times$ 256 and used it as a target. The target is downsampled (4×4) times by the bicubic interpolation, to 64 $\times$ 64 low resolution input image. Because our patch size is 4 by 4, the label size is 64 $\times$ 64. the classifier receives 64 $\times$ 64 low resolution image as an input so that the classifier doesn't need to downsize the feature map. The number of classes is set to 7, which is the same as the networks for compression artifact removal. For encoder and decoder network we adopted the architecture SRResNet [1] and used it for baseline. Figure2 shows our super resolution result. as the result shown our method also works well with super resolution.

References

[1] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint arXiv:1609.04802*, 2016.

[2] H. R. Sheikh, Z. Wang, L. Cormack, and A. C. Bovik. Live image quality assessment database release 2 (2005), 2016.