NTIRE 2022 Efficient SR Challenge Factsheet -title of the contribution-

Wang WanJun
School of Control and Computer Engineering, North China Electric Power University
2 Beinong Road, Changping District, Beijing
1206371055@qq.com

Zhou DengWen
School of Control and Computer Engineering, North China Electric Power University
2 Beinong Road, Changping District, Beijing
1206371055@qq.com

1. Team details

- ncepu_explorers
- Wang WanJun
- 2 Beinong Road, Changping District, Beijing, 18800157597, 1206371055@qq.com
- Zhou DenWen,Liu YuKai
- Team website URL (if any)
- North China Electric Power University
- The team and / or team members have no relationship with ntire 2022 sponsors
-
- 28.79dB
- testing code: https://github.com/wwjfsfs/wwjyyds/tree/main; training code: https://github.com/wwjfsfs/MDAN

2. Method details

Our proposed MDAN network architecture, shown in Figure 1 (a), consists of four main parts: shallow feature extraction block (SFEB), nonlinear feature mapping block (NFMB), hierarchical feature fusion block (HFFB), and upsampling block (Upsampler). The SFEB consists of only one 3×3 convolutional layer and one leaky rectified linear unit (LReLU), and the Upsampler uses the sub-pixel convolution. The NFMB cascades N (in our experiments, N = 3) area feature fusion blocks (AFFBs). The HFFB mainly consists of multiple pairs of the lightweight convolutional units (LConvSs) / 1×1 convolutions and multiple multi-dimensional attention blocks (MDABs).

Given the input LR image I_{LR}, it is first input to the SFEB to extract the shallow feature F_0:

$$F_0 = LReLU\left(C_{3\times3}\left(I_{LR}\right)\right) \tag{1}$$

where $C_{3\times3}(\cdot)$ is the 3×3 convolution function, and $LReLU(\cdot)$ is the LReLU activation function. F_0 is then input into the N cascaded AFFBs in the NFMB to extract the deep features:

$$F_n = f_{AFFB}^n\left(F_{n-1}\right)(n = 1, 2, \cdots, N) \tag{2}$$

where $f_{AFFB}^n(\cdot)$ is the n-th AFFB block function, F_{n-1} is the output feature of the $(n-1)$-th AFFB, i.e., the input feature of the n-th AFFB, and the output feature of each AFFB is input to the HFFB for the hierarchical feature information fusion:

$$F_{fusion} = f_{HFFB}\left(F_1, F_2, \cdots, F_N\right) \tag{3}$$

where $f_{HFFB}(\cdot)$ is the hierarchical feature fusion function, and F_{fusion} is its output. The sum of F_{fusion} and F_0 is input to the Upsampler to obtain the SR residual image, and the target SR image is:

$$I_{SR} = f_{Up}\left(F_{fusion} + F_0\right) + f_{bic}\left(I_{LR}\right) \tag{4}$$

where $f_{Up}(\cdot)$ is the Upsampler function, and $f_{bic}(\cdot)$ is the bicubic upsampling function.

2.1. Area Feature Fusion Block (AFFB)

As shown in Figure 1 (b), the AFFB consists of two branches, from top to bottom, Branch 1 and Branch 2.

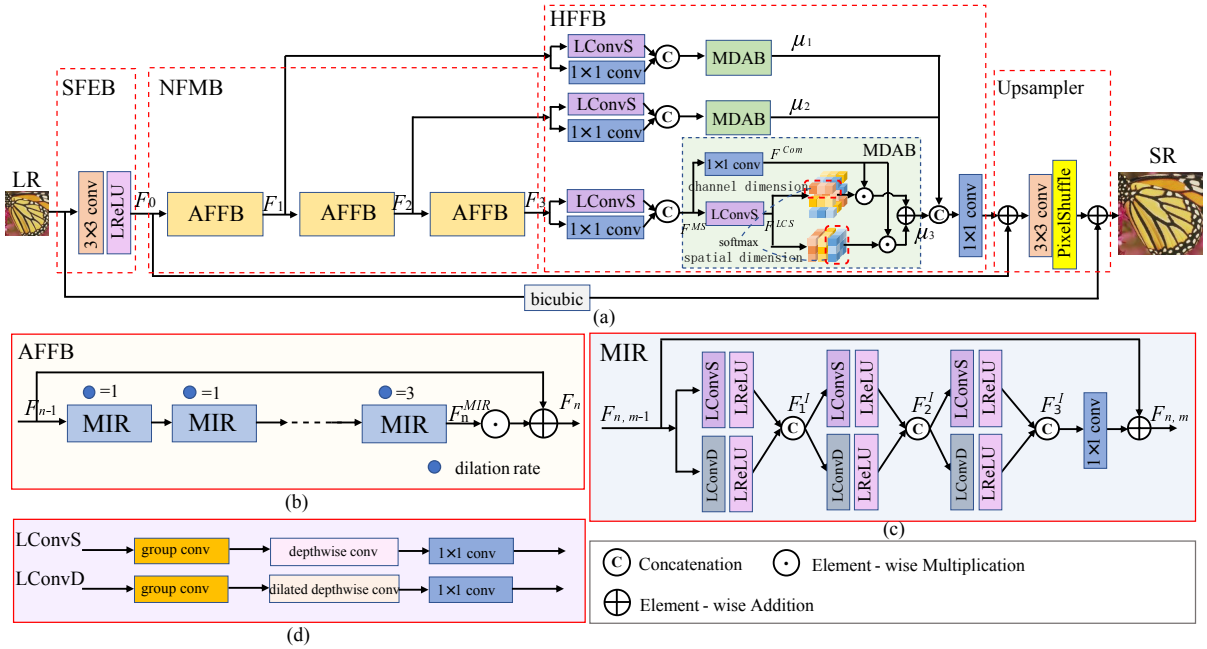

Figure 1 (a)The network architecture of our proposed MDAN. (b) Area feature fusion block (AFFB). (c) Multiple interactive residual block (MIR). (d) Lightweight convolutional units (LConvS / LConvD).

Branch 1 is just a simple residual connection; Branch 2 contains M (in our experiments, M = 6) cascaded multi-interaction residual blocks (MIRs). The dilation convolutions in the MIRs (please see Section 3.1.2) may have different dilation rates, e.g., which are 1, 1, 2, 2, 3, 3 for the six cascaded MIRs to capture the contextual features at different scales.

The output feature of the Branch 2 in the n-th AFFB can be expressed as:

$$F_n^{MIR} = f_{MIR}^{M}\left(\cdots f_{MIR}^{2}\left(f_{MIR}^{1}\left(F_{n-1}\right)\right)\right) \quad (5)$$

where $f_{MIR}^{m}(\cdot)$ is the m-th (m = 1, 2, ... , M) MIR function in the n-th (n = 1, 2, ..., N) AFFB. F_{n-1} is the output feature of the $(n-1)$-th AFFB, i.e., the input feature of the n-th AFFB.

The output features of the n-th AFFB can be expressed as:

$$F_n = F_n^{MIR} + F_{n-1} \quad (6)$$

2.1.1 Multiple Interactive Residual Block (MIR)

Li et al. propose a dual-branch multi-scale residual block (MSRB) with one branch using two 3 × 3 convolutional kernels and the other branch using two 5 × 5 convolutional kernels to capture the multi-scale features, and perform the cross-scale feature interactions. The MSRB is effective, and our MIR is similar to the MSRB, as in Figure 1 (c). However, the MSRB is not lightweight enough. We design two very lightweight convolutional units, called LConvS and LConvD, as in Figure 2 (d). The LConvS consists of a 1 × 1 group convolution (our experiments use 3 groups), a 3 × 3 depth-wise convolution, and a 1 × 1 convolution. The only difference between LConvD and LConvS is that the depth-wise convolution in LConvS is replaced by a 3 × 3 depth-wise dilated convolution. We replace the 3 × 3 convolutions in the MSRB with LConvS and the 5 × 5 convolutions in the MSRB with LConvD, which can significantly reduce the number of parameters. In addition, each branch of the MSRB has only two convolutional layers. Our experiments show that increasing the number of the convolutional layers and the number of interactions of the cross-scale features can improve performance. We carry out three interactions of the cross-scale features (the MSRB interacts only twice). Our MIR is also more flexible than the MSRB because it is easier to obtain more multi-scale features by adjusting the dilation rates of the depth-wise dilated convolutions in LConvD. The output of the first multi-scale feature interaction of the m-th MIR in the n-th AFFB can be expressed as (ignoring the LReLU nonlinear activation):

$$F_1^I = \left[f_{LCS}\left(F_{n,m-1}\right), f_{LCD}\left(F_{n,m-1}\right)\right] \quad (7)$$

where $f_{LCS}(\cdot)$ is the LConvS function, $f_{LCD}(\cdot)$ is the LConvD function (its dilation rate may be 1,2, or 3), $[\cdot]$ is the concatenation operation, and $F_{n,m-1}$ is the output of the (m-1)-th MIR in the n-th AFFB, i.e., the input of the m-th MIR in the n-th AFFB. The calculation of the output

F_2^I for the second interaction is similar to Eq. (14), except that $F_{n,m-1}$ is replaced by F_1^I. The calculation of the output F_3^I for the third interaction is also similar to that of F_2^I. The output feature $F_{n,m}$ of the m-th MIR in the n-th AFFB block are:

$$F_{n,m} = C_{1\times1}\left(F_3^I\right) + F_{n,m-1} \tag{8}$$

where $C_{1\times1}(\cdot)$ is the 1×1 convolution function.

2.2. Hierarchical Feature Fusion Block (HFFB)

The output feature of each AFFB in the NFMB is input to the HFFB for the further hierarchical feature information fusion. The HFFB has N main branches, and each main branch processes the output of a corresponding AFFB, as shown in Figure 1 (a). The output of each AFFB is processed by a 1×1 convolution subbranch and an LConvS subbranch in the main branch, respectively, and then the output features of the two subbranches are concatenated in the channel dimension, and input to the MDAB. The outputs of the N major branches are adaptively concatenated in the channel dimension, and the concatenated feature is again passed through a 1×1 convolution to obtain the output of the HFFB.

$$F_{\text{fusion}} = C_{1\times1}(\sum_{n=1}^{N}(\mu_n(f_{MDAB}^n([C_{1\times1}(F_n), f_{LCS}(F_n)])))) \tag{9}$$

where $C_{1\times1}(\cdot)$ is the 1×1 convolution function, $f_{LCS}(\cdot)$ is the LConvS function, $[\cdot]$ is the channel concatenation operation, $f_{MDAB}^n(\cdot)$ is the n-th MDAB function, $\mu_n\,(n = 1, 2, \cdots, N)$ is the learnable scalar weight, F_n is the output of the n-th AFFB, and F_{fusion} is the output of the HFFB.

2.2.1 Multi-dimensional Attention Block (MDAB)

In current SISR models, both the channel attention and the spatial attention mechanisms are widely used. Assuming that the size of the feature map is $C \times H \times W$ (C, H and W are the number of channels, height and width of the feature, respectively), the channel attention computes a one-dimensional vector (i.e., weights) ($C \times 1 \times 1$), modeling the dependencies between the channels, while spatial attention computes a two-dimensional matrix (i.e., weights) ($1 \times H \times W$) that models the dependencies between spatial locations. The first-order triplet proposed by Zhang et al. is similar to channel attention, which can capture cross-dimension interaction between the (C, H), (C, W), and (H, W) dimensions. The pixel attention (PA) proposed by Zhao et al. uses a 1×1 convolution and a Sigmoid function to compute a three-dimensional pixel-wise matrix ($C \times H \times W$), which is similar channel attention and spatial attention. Our MDAB is shown in Figure 1 (a), which is similar to the PA, also computes a three-dimensional pixel-wise matrix ($C \times H \times W$). The differences are : (1) The 1×1 convolution is replaced by the LConvS block, and the Sigmoid function is replaced by the Softmax function. (2) We learn the dependencies between the pixels of the feature in the channel dimension and the spatial dimension, respectively.

The input feature of the MDAB is assumed to be F^{MS}, which passes through a 1×1 convolution and an LConvS, respectively. The output of the LConvS passes through the Softmax function in the channel dimension and the space dimension, respectively, to obtain two pixel-level attention weights. The output of the 1×1 convolution is weighted by the two attention weights, and summed. After F^{MS} passes through the 1×1 convolution, the output feature can be expressed as:

$$F^{Com} = C_{1\times1}\left(F^{MS}\right) \tag{10}$$

where $C_{1\times1}(\cdot)$ is the 1×1 convolution function, and F^{Com} is its output. After F^{MS} passes through the LConvS, the output feature can be expressed as:

$$F^{LCS} = f_{LCS}\left(F^{MS}\right) \tag{11}$$

where $f_{LCS}(\cdot)$ is the LConvS function, and F^{LCS} is its output. The output of the MDAB can be expressed as:

$$F^{MDA} = F^{Com} \odot \tau_1\left(F^{LCS}\right) + F^{Com} \odot \tau_2\left(F^{LCS}\right) \tag{12}$$

where $\odot$ is the element-wise multiplication operation, $\tau_1(\cdot)$ refers to the operation of the Softmax function acting on the channel dimension of the feature, $\tau_2(\cdot)$ refers to the operation of the Softmax function acting on the spatial dimension of the feature, and F^{MDA} is the output feature of the MDAB.Our MDAB is simple, effective, which can also be easily embedded into other SR models.

3. training strategy

We used the DIV2K(001-800) and Flicker2K dataset as the training data, and the 100 images (801-900) of DIV2K were used for validation. The original HR training images were bicubically downsampled to obtain the paired L-R training images. Similar to other methods, the training images were augmented by randomly rotating and flipping horizontally. In the model training, we randomly cropped 64 patches of size 64×64 from the LR images as input for each training minibatch. The ADAM optimizer was used with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 10^{-8}$. Our $2\times$ model was trained for 1000 epochs with an initial learning rate set to 2.5×10^{-3} and halved for every 200 epochs, and training data is only the DIV2K(001-800). Based on the $2\times$ model, the $4\times$ SR models were trained 500 epochs, and the training data are DIV2K(001-800) and Flicker2K dataset. The

L1 loss function was used. The model was implemented by using PyTorch with a NVIDIA RTX 2080Ti GPU.

In our MDAN architecture, three AFFBs were cascaded in the NFMB, and the number of input and output channels for each AFFB was 48. Six MIRs were cascaded in each AFFB, and the dilation rates of the dilation convolutions in each MIR were set to 1, 1, 2, 2, 3 and 3. In each MIR, the number of the input channels of both LConvS and LConvD was 48 and the number of the output channels was 24. The group convolutions in both LConvS and LConvD used three groups. The initial values of the learnable parameters μ_1, μ_2, and μ_3 in the HFFB were set to 0.3, 0.3, and 0.4, respectively.

4. Quantitative comparisons

For example, RFDN had 40% more parameters and 40% more computations than our MDAN, but the PSNR/SSIM results of our MDAN were consistently better than those of RFDN, with a maximum PSNR difference of 0.13dB. LAPAR-A had 70% more parameters and nearly 4 times more computations than our MDAN, but the PSNR/SSIM results of our MDAN were also consistently better than those of LAPAR-A.