Reparameterized Residual Feature Network For Lightweight Image Super-Resolution

Weijian Deng1, Hongjie Yuan1, Lunhui Deng1, Zengtong Lu2
1Communication University of China, China
2Ruijie Networks Co., Ltd. China
{348957269, homjayuan}@qq.com, dy3000@cuc.edu.cn, luztxp@gmail.com

Abstract

In order to solve the problem of deploying super-resolution technology on resource-limited devices, this paper explores the differences in performance and efficiency between information distillation mechanism and residual learning mechanism used in lightweight super-resolution, and proposes a lightweight super-resolution network structure based on reparameterization, named RepRFN, which can effectively reduce GPU memory consumption and improve inference speed. A multi-scale feature fusion structure is designed so that the network can learn and integrate features of various scales and high-frequency edges. We rethought the redundancy existing in the overall network framework, and removed some redundant modules without affecting the overall performance as much as possible to further reduce the complexity of the model. In addition, we introduced a loss function based on Fourier transform to transform the spatial domain of the image into the frequency domain, so that the network can supervise and learn the frequency part of the image. The experimental results show that the RepRFN designed in this paper achieves relatively low complexity while ensuring certain performance, which is conducive to the deployment of Edge devices. Code is available at https://github.com/laonafahaodange/RepRFN.

1. Introduction

Super-resolution (SR) is an important branch of image reconstruction in computer vision and a hot research topic in recent years. It is widely used in the fields of medical treatment, security, image and video reconstruction, and even game image enhancement. In recent years, many SR networks based on convolutional neural network (CNN) have been proposed, indicating that CNN plays a role in promoting the development of image SR.

In 2014, Dong et al. applied convolutional neural network (CNN) to the super-resolution problem for the first time and proposed SRCNN [5]. Compared with traditional methods, SRCNN achieved good results with only three convolution layers, proving the effectiveness of deep learning in this problem. Kim et al. proposed a deeper SR network VDSR [10] by cascading multiple small size convolution to ensure the same receptive field while reducing parameters. The EDSR [17] proposed by Lim et al. further improves the depth by the residual structure and multi-scale technology, achieving better results than SRCNN. This shows that the depth of the network is an important factor affecting the quality of SR image reconstruction, and constructing a deeper network can improve the performance of SR.

However, most SR networks sacrifice the efficiency in order to improve the details of image restoration. In some
scenarios, real-time performance will also affect user experience. Therefore, how to efficiently extract image edge, texture, structure and other information, and balance the relationship between the performance and complexity of SR network is a key research, which determines whether the network can be deployed on devices with limited resources such as computing and storage units. To this end, we propose a novel Reparameterized Residual Feature Network, referred to as RepRFN. A multi-branch structure is designed to extract features of different receptive fields by using multiple parallel convolution kernels of different sizes, and feature fusion is realized by local residual connection. In order to extract the edge information effectively, the Sobel branch and Laplace branch in the Edge-oriented Convolution Block (ECB) [29] are introduced into the multi-branch structure. In the training stage, we regard the SR task as a multi-task learning problem of spatial domain learning and frequency domain learning. Fourier transform is introduced into the loss function to calculate the loss in the frequency domain to guide the model to recover frequency information. Experiments show that the proposed RepRFN achieves a balance in performance and efficiency.

Our contributions can be summarized as follows:

1. A multi-scale feature fusion structure based on reparameterization is proposed. Features are convolved by multiple parallel convolution of different receptive field and edge-oriented convolution modules to extract features of different modes. Residual connection is used to aggregate features, which improves the expression ability of features.

2. The structure of RFDN [18] model was reconsidered, we analyzed the redundancy of RFDN, and the 1 × 1 convolution used for channel transformation was removed in our network.

3. Fourier transform is introduced into the loss function, so that the model can learn the frequency information of the image in the process of supervised training, and enhance the recovery ability of frequency details.

2. Related work

The SR network SRCNN [5] has achieved impressive results, but there are some problems such as large computation. Dong et al. achieved a learning upsampling by removing interpolation upsampling, introducing transposed convolution at the end of the network, and using smaller but more convolution kernels for feature extraction. Based on these improvements, they proposed the lightweight SR network FSRCNN [6], which achieved about 17 times of acceleration compared with SRCNN. Kim et al. proposed a deep recursive convolutional network DRCN [11] by recursively invoking the feature extraction layer. DRRN [22] improves DRCN by combining recursion and residual network to achieve better performance with fewer parameters. NamhyukAhn et al. adjusted model efficiency using group convolution, they adopted a mechanism similar to recursive network to share parameters among cascade modules, and proposed a lightweight cascade residual network CARN [1]. Lai et al. proposed LapSRN [13], they removed the preprocessing step of bicubic interpolation of input, transposed convolution is used for upsampling, with low resolution (LR) image as input, feature maps are extracted through the cascade convolutional layer, then a convolution layer is used to learn the residual difference between high-resolution image and up-sampled feature image to complete first-level reconstruction, and multi-scale reconstruction is finally realized through stepwise upsampling. Hui et al. proposed an Information Distillation Network IDN [9]. The key to IDN lies in the information distillation module. Each information distillation module contains an enhancement unit and a compression unit, which can effectively extract local long and short-path features, in addition, IDN uses relatively few convolution kernels and group convolution, so the inference speed is relatively fast. On the basis of IDN, Information Multi-Distillation Network IMDN [8] constructs a cascable Information Multi-Distillation Block IMDB, which consists of distillation and selective fusion, specifically, the distillation module gradually extracted features, and the fusion module determined the importance of candidate features according to the attention mechanism and fused them. IMDN won the first prize in AIM2019 Challenge on Constrained Super-Resolution [27]. Subsequently, Liu et al. reconsidered IMDN and proposed Residual Feature Distillation Network RFDN [18], they think that the key component of both IDN and IMDN is the information distillation mechanism IDM, which explicitly divides extracted features into two parts, one retained and the other further extracted. However, the efficiency of this mechanism is still not enough to integrate the residual connection with IDM, therefore, they designed a Shallow Residual Block SRB as the main module of RFDN, so that the network can achieve lightweight and maximize the use of residual learning. Its E-RFDN won first place in AIM2020 Challenge on Efficient Super-Resolution [26].

3. Method

In view of the excellent performance and efficiency of information distillation mechanism in efficient super-resolution challenge in recent years [26, 27], we focuses on the comparison of information distillation mechanism. The residual network is widely used by researchers because of its simple structure, easy implementation and good optimization effect. Since there is no need to concatenate channels, the inference speed is faster and the operation is more efficient, and there is greater potential for deploying
on Edge device with limited resources.

In Section 3.1, we propose a Residual Feature Network (RFN) for lightweight image SR. Compared with the information distillation mechanism, we observe the differences in performance and efficiency between the residual feature learning mechanism and the information distillation mechanism through experiments. In Section 3.2, we review the shortcomings of RFN, make a series of improvements to the model, and propose a multi-scale feature fusion lightweight SR network RepRFN based on Reparameterization [3, 4, 29]. In Section 3.3, we describe a loss function based on Fourier transform, which converts images from spatial domain to frequency domain, so that the model can learn frequency information in the process of training, and improve the performance of lightweight SR model.

3.1. Residual Feature Network (RFN)

The overall network structure is shown in Figure 2. The network consists of three main parts: Shallow Feature-extraction Block (SFB), Deep Feature-extraction Block (DFB) and Upsample Block (UB). The SFB is used to extract the shallow features of the input LR image and DFB carries out further nonlinear mapping on the extracted shallow features to obtain deeper feature expression, then deep feature and shallow feature are fused through residual connection to obtain fusion features. Finally, the UB performs pixel recombination on the fused features to reconstruct a SR image.

The SFB consists of a $3 \times 3$ convolution layer, which is mainly responsible for extracting shallow feature from the input LR image. The DFB is composed of stacked Residual Feature Block (RFB), which gradually extracts shallow features and uses residual learning to integrate shallow features and deep features to improve the expression ability of features. The UB is the subpixel convolution layer, which is composed of a $3 \times 3$ convolution layer and a PixelShuffle layer [21]. The final SR image is obtained by recombining the fused features and pixel mapping. The above process can be expressed as:

$$X_{sf} = F_{sf}(I_{lr})$$
$$X_{df}^i = F_{df}^{i-1}, i = 2, 3, 4, 5$$
$$X_1 = X_{sf}$$
$$X_{fusion} = conv_{3x3}(X_{df}^5) + X_{sf}$$
$$I_{sr} = Up(X_{fusion})$$

where $I_{lr}$ represents the input LR image, and shallow features $X_{sf}$ are obtained after SFB $F_{sf}$. Then the shallow features obtained are input into the $i$-th DFB $F_{df}^{i-1}$ of the stacked RFBs, and the $i$-th deep features $X_{df}^i$ are extracted layer by layer. After $3 \times 3$ convolution, the extracted deep feature is fused with the shallow feature to obtain the fused features $X_{fusion}$. Finally, the fused features are input into the UB $Up(\cdot)$ to obtain the reconstructed SR image $I_{sr}$.

Residual Feature Block The key of the Residual Feature Block lies in the residual feature learning mechanism.
Different from the information distillation mechanism, the information distillation mechanism divides the input feature into two parts along the channel dimension. One part is retained and the other part is input to the next information distillation module for further feature extraction. After several distillation steps, the feature fusion is completed by concatenating along the channel dimension, so as to realize the fusion of distillation information. However, the residual feature learning mechanism does not split the features along the channel dimension, but directly inputs the extracted features into the next part, and only adds and merges the extracted deep features and shallow features in each module, which alleviates the problems of large GPU memory consumption and increased inference time caused by the channel split and concatenation operation. Figure 3 shows several information distillation module. Figure 3a, Figure 3b and Figure 3c respectively represent the structure of Information Distillation Block (IDB) used in IDN [9], IMDN [8] and RFDN [18]. The RFB (Figure 3d) used in this section draws on the structure of RFDN-IDB. Specifically, the difference lies in that the input features are no longer operated by channel split, but directly input to the next convolution layer, and the information fusion mechanism is replaced by residual fusion.

Assume \( f_{k=n} \) represents the output feature of the \( i \)-th \( n \times n \) convolution in the RFB, \( \text{conv}_{n \times n} \) represents the operation function of the \( i \)-th \( n \times n \) convolution layer that has been continuously stacked. \( f_{\text{fusion}} \) represents the fusion feature generated by residual connection between input feature \( x_{in} \) and intermediate feature \( f_{k=3} \). Attention represents the attention mechanism used by the module, and \( x_{out} \) represents the output feature. The calculation process of this module can be expressed as:

\[
\begin{align*}
 f_{k=3}^4 &= \text{conv}_{3 \times 3}^4(x_{in}) \\
 f_{\text{fusion}} &= f_{k=3}^4 + x_{in} \\
 f_{k=1}^1 &= \text{conv}_{1 \times 1}^1(f_{\text{fusion}}) \\
 x_{out} &= \text{Attention}(f_{k=1}^1)
\end{align*}
\]  

(2)

We explored the performance and efficiency differences between the information distillation mechanism and the residual learning mechanism. RFDN [18] was used as the representative of information distillation mechanism. It is noted that global residual connections are used in RFDNs, and the impact of different residual connections on performance differences was also explored. As shown in Figure 4, RFB1, RFB2 and RFB3 are defined to represent different residual connection modes: local residual connection, global residual connection, and local combined global residual connection. The attention mechanism uses the same Enhanced Spatial Attention (ESA) as RFDN, we used the plane model without any residual connection as the baseline model. The number of output channels of each module was unified as 48, and RFDN was retrained with the above training setting, which is denoted as RFDN*.

As shown in the Table 1, although the network structure based on residual feature learning has a slight increase in the number of model parameters and the amount of computation compared with RFDN, it has a decrease in inference time, number of activations, number of convolution layers, and maximum GPU memory consumed during inference. In particular, Mem decreased by 54.5%. It shows that the residual feature learning mechanism reduces the GPU memory consumed and time cost caused by the channel split and concatenation operation compared with the information distillation mechanism. By comparing different residual connection, it can be seen from the data of RFB1, RFB2 and RFB3 that RFB2 using global residual connection is slightly lower than RFB1 using local residual connection in terms of PSNR when the number of parameters and calculation amount are approximately the same, and RFB1 using only local residual connection has the same performance as RFB3 combined with local residual connection and global residual connection. It can be known that the contribution of global residual connection is lower than that of local residual connection.

### 3.2. Reparameterized Residual Feature Network (RepRFN)

Despite the low GPU memory usage and fast inference speed of RFN, there are still many problems. From the feature scale of the model, 3 × 3 convolution layer is mostly...
used in feature extraction, and the receptive field is relatively simple. Secondly, the structure of the model is still redundant. In addition, the extraction and recovery of high-frequency information in the image feature domain are also deficient. To solve the above problems, this section makes a series of improvements to the model, and proposes a multi-scale feature fusion lightweight SR network RepRFN based on Reparameterization [3, 4, 29]. To solve the problem of simple receptive field of the model, multiple parallel branch structure was designed, and the features of different receptive fields and modes are extracted and fused to make the model benefit from the multi-branch structure as much as possible. At the same time, the reparameterization, decoupling training and inference process are introduced to avoid the problem that the number of parameters and calculation amount increase caused by the introduction of multi-branch structure. To solve the problem of model structure redundancy, we reconsidered and analyzed the structural differences between RFN and RFDN, removes the redundancy, we reconsidered and analyzed the structural differences between RFN and RFDN, the 1 × 1 convolution layer used for channel transformation in RFN and makes structural improvements to ESA.

The RepRFN network structure proposed in this paper is shown in Figure 2. RepRFN has the same structure as RFN, the difference is that RepRFN replaces RFBs in RFN with Reparameterized Residual Feature Blocks (RepRFBs). Reparameterized Block (RepBlock) is the main component of RepRFB, and multi-branch structure constitutes RepBlock as shown in Figure 6. Shallow features are gradually extracted from different patterns of features by stacked RepBlocks in RepRFBs and fused through residual connections. Then deep features are obtained through 3 × 3 convolution layer. Then Shallow features and deep features are fused by local residual connection to improve the expression ability of features. The upsampling module uses subpixel convolution [21] to generate the final SR image. The above process can be expressed similarly as Equation 1.

Reparameterized Residual Feature Block The design of RepRFB proposed in this paper refers to the structure of RFDB in RFDN. In RFDB, the intermediate feature is split three times by the SRB and 1 × 1 convolution layer in each information distillation module as shown in Figure 3c. Therefore, the first three layers of RepRFB adopt a reparameterized multi-branch structure, which is called RepBlock in this paper, features propagate through multiple paths that perform different operations and eventually fuse them to improve the expressiveness of the model. In RFDB, due to the channel concatenation operation, a channel transformation using 1 × 1 convolution is required after the concatenation operation to input to the attention layer. However, in RepRFB, due to the existence of local residual connections, the size and number of channels of intermediate features before and after the RepBlock and convolutional layer are always unchanged, so channel transformation operation is not required. Therefore, it can be considered that the 1 × 1 convolution in RepRFB is redundant, and the 1 × 1 convolution is removed to further compress the parameters. Kong et al. [12] analyzed the redundancy of ESA in RFDN based on the pruning sensitivity analysis tool of the one-shot structured pruning algorithm [15], and found that the three convolution layers in the convolution group ranked first, third and fourth respectively in terms of redundancy. Therefore, the three-layer convolution of the convolution group in each ESA was reduced to one layer. The final RepRFB structure is shown in Figure 6.

Assume $f_{ms}^i$ represents the output feature of the $i$-th multi-branch structure in the RepRFB, $\text{repblock}^3(\cdot)$ represents the operation function of the $i$-th multi-branch structure that has been continuously stacked, $f_{fusion}$ represents the fusion feature generated through residual connection between input feature $x_{in}$ and intermediate feature $f_{c3}$. $\text{Attention}$ represents the attention layer used by the module, and $x_{out}$ represents the output feature, it can be expressed as:

$$
\begin{align*}
    f_{ms}^3 &= \text{repblock}^3(x_{in}) \\
    f_{c3} &= \text{conv}^{3 \times 3}(f_{ms}^3) \\
    f_{fusion} &= f_{c3} + x_{in} \\
    x_{out} &= \text{Attention}(f_{fusion})
\end{align*}
$$

3.3. Loss Function Based on Fourier Transform

For the problem of extracting and restoring high-frequency information, in addition to introducing ECB [29] into multi-branch structure, Fourier transform is introduced into loss function to guide the model to learn frequency domain features and restore high-frequency information as much as possible.

The commonly used loss function of SR include L1 loss, L2 loss and Charbonnier loss [13], etc. These loss functions can be considered as a kind of pixel loss by measuring the difference between the pixel values of SR images and high resolution (HR) images to guide model learning. How to effectively restore high-frequency information of images in SR has always been a focus of industry attention. During the model training process, the learning of frequency information is achieved by measuring the pixel differences between spectral maps of SR and HR. As expressed in the Equation 4:

$$
\mathcal{L}(x, y) = \mathcal{L}_{pix}(\text{fft}(x), \text{fft}(y))
$$

4. Experiments

4.1. Experimental Setup

In terms of training, DIV2K [24] training set and Flickr2K [24] dataset are used, the HR patches is set to 192 × 192, random horizontal flip, vertical flip and rotation
are introduced into the data augmentation during training. The proposed RepRFN consists of 4 RepRFBs, the number of channels is set to 48. The model is trained from scratch. We use the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$, batchsize is set to 64, and the initial learning rate is set to $5 \times 10^{-4}$ and halved at every 100 epochs. The total number of epochs is 1001. In the process of training, the loss function used is the combination of pixel loss and loss function based on Fourier transform. In practical application, Charbonnier loss [13] can avoid the problem that the results generated by L1 loss and L2 loss are too smooth [30], in the experiment, we also found that the Charbonnier loss is better than the L1 loss in terms of PSNR, so we choose the Charbonnier loss as $L_{pix}$. It should be noted that we only perform Fourier transform on the scale dimension of the image. Finally, the loss can be formulated as:

$$\mathcal{L}(x, y) = \lambda_1 L_{pix}(x, y) + \lambda_2 L_1(\text{fft}(x), \text{fft}(y))$$  \hspace{1cm} (5)$$

where $\lambda_1 = 0.9$, $\lambda_2 = 0.1$ and the hyperparameter $\epsilon^2$ in Charbonnier loss is set to $10^{-6}$.

### 4.2. Quantitative Results

We compared several SR networks based on CNN [1, 5, 6, 8–11, 14, 18, 22, 23, 28, 29]. The PSNR and SSIM of the model were tested on five benchmark datasets [2, 7, 19, 20, 25] to measure model performance. PSNR and SSIM were tested on the Y-channel in the YCbCr color space of the image. In terms of model complexity, assuming that the model output is a 720P image, the Parameters and FLOPs are used to measure the model complexity. In order to maximize the potential performance of the RepRFN proposed in this paper, Geometric Self-ensemble [17] was used in the experiment, and the RepRFN using the Geometric Self-ensemble strategy is denoted as RepRFN+. The experimental results are shown in Table 2. Figure 1 compares the PSNR and Parameters of different networks on the Set14 [25] under the condition of a scale factor of 4. It can be seen that RepRFN achieves performance comparable to other networks with fewer parameters and computational complexity, achieving a better balance in performance and efficiency. The network visualization results are shown in the Figure 7.

In order to further validate the efficiency of model deployment on mobile devices, we also compared the differences in inference time among three popular model deployment schemes on the Android devices of Qualcomm Snapdragon 865 and 820, and Rockchips RK3588 hardware platforms. The three deployment schemes are ONNX, PaddleLite, and Rockchips RKNN. As shown in Table 6, the efficiency of the RepRFN has been verified through experiments, indicating that the proposed RepRFN has certain competitiveness in deployment on hardware platforms.

![Figure 7. Visualization results.](image-url)
Table 2. PSNR/SSIM and complexity results of SR with different scale factors for different networks on different benchmark test sets. The best and second-best results are marked in red and blue colors, respectively.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RepRFN-P</td>
<td>2</td>
<td>32.18/0.9604</td>
<td>33.39/0.9197</td>
<td>32.48/0.9269</td>
<td>32.89/0.9303</td>
<td>38.80/0.9774</td>
<td>57</td>
<td>11113.48</td>
</tr>
<tr>
<td>RepRFN+ (ours)</td>
<td>3</td>
<td>32.18/0.9604</td>
<td>33.39/0.9197</td>
<td>32.48/0.9269</td>
<td>32.89/0.9303</td>
<td>38.80/0.9774</td>
<td>57</td>
<td>11113.48</td>
</tr>
<tr>
<td>RepRFN+ (ours)</td>
<td>4</td>
<td>32.18/0.9605</td>
<td>33.39/0.9197</td>
<td>32.48/0.9269</td>
<td>32.89/0.9303</td>
<td>38.80/0.9774</td>
<td>57</td>
<td>11113.48</td>
</tr>
</tbody>
</table>

Table 3. Performance differences between RepRFN-P and RepRFN on different benchmark datasets (Y-channel in Ycbcr color space).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RepRFN-P</td>
<td>2</td>
<td>32.18/0.9604</td>
<td>33.39/0.9197</td>
<td>32.48/0.9269</td>
<td>32.89/0.9303</td>
<td>38.80/0.9774</td>
<td>57</td>
<td>11113.48</td>
</tr>
<tr>
<td>RepRFN+ (ours)</td>
<td>3</td>
<td>32.18/0.9604</td>
<td>33.39/0.9197</td>
<td>32.48/0.9269</td>
<td>32.89/0.9303</td>
<td>38.80/0.9774</td>
<td>57</td>
<td>11113.48</td>
</tr>
<tr>
<td>RepRFN+ (ours)</td>
<td>4</td>
<td>32.18/0.9605</td>
<td>33.39/0.9197</td>
<td>32.48/0.9269</td>
<td>32.89/0.9303</td>
<td>38.80/0.9774</td>
<td>57</td>
<td>11113.48</td>
</tr>
</tbody>
</table>

Table 4. Comparative experiment on model structure redundancy (DIV2K [24] validation set, RGB color space).

<table>
<thead>
<tr>
<th>Channels</th>
<th>Remove 1 × 1 Conv</th>
<th>Modify ESA</th>
<th>Param Reduction (%)</th>
<th>FLOPs Reduction (%)</th>
<th>PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>48</td>
<td>√</td>
<td>×</td>
<td>0.429</td>
<td>22.68</td>
<td>28.94</td>
</tr>
<tr>
<td>48</td>
<td>√</td>
<td>×</td>
<td>0.411</td>
<td>22.66</td>
<td>28.97</td>
</tr>
<tr>
<td>50</td>
<td>×</td>
<td>×</td>
<td>0.433</td>
<td>23.50</td>
<td>28.99</td>
</tr>
</tbody>
</table>

4.3. Ablation Study

Multi-Scale Feature Fusion Module In order to verify the impact of multi-branch structure on model performance,
we designed an experiment, which referred to the reparameterized planar model of RepRFN proposed in this paper as RepRFN-P (P: Plain), meaning that RepRFN-P does not contain multi-branch structure, similar to the RFN structure proposed in Section 3.1. In terms of experimental setup, it is the same as RepRFN, and RepRFN-P was retrained to observe the performance differences between it and RepRFN. From the Table 3, it can be seen that RepRFN is relatively superior in performance to RepRFN-P without using a multi-branch structure, indicating that the multi-branch structure is beneficial for improving the performance of the model.

**Model Structure** We explored the redundancy of the model structure. Starting from three aspects: the number of model channels, $1 \times 1$ convolution used for channel transformation, and ESA. We investigated the impact of these three factors on model performance and efficiency. The baseline model is based on a model with a channel number of 48, preserving $1 \times 1$ convolution and the original ESA module. From the Table 4, it can be seen that the model performance of the modified ESA module has been increased by about 0.03dB, indicating that there is some redundancy in the convolution groups in ESA module. As the number of channels increases to 50, the PSNR also correspondingly increases by 0.02dB, with an 8% increase in both parameter and FLOPs. When the number of channels increases to 52, the PSNR no longer increases and the model complexity increases. This indicates that the impact of channel numbers on model performance is manifested as the larger the number of channels, the more saturated the model performance tends to be, and the higher the complexity. Under the modified ESA module with the same number of channels of 50, removing the $1 \times 1$ convolution used for channel transformation can reduce the amount of parameters and FLOPs, but the performance also decreases by about 0.04dB. In order to obtain a model with lower complexity, we sacrificed some performance to design a model with lower complexity, the model RepRFN ultimately adopts the design of channel number 48, modified ESA module, and removes $1 \times 1$ convolutions used for channel transformation.

**Loss function** In order to verify the effectiveness of the proposed Fourier transform based loss function, we explored the impact of L1 loss, Charbonnier loss, and Fourier transform based loss on the performance of the SR model. It can be seen from the Table 5 that Charbonnier loss is better than L1 loss in PSNR. After the introduction of Fourier transform based loss function, the performance of the model has been increased, indicating that the Fourier transform based loss function is beneficial to the model performance.

### 4.4. NTIRE 2023 ESR challenge

We have participated in NTIRE 2023 Efficient Super-Resolution Challenge [16]. This competition aims to devise a network that reduces one or several aspects such as runtime, parameters, FLOPs, activations, and depth of RFDN while at least maintaining PSNR of 29.00dB on validation datasets. Our results are shown in the Table 7.

### 5. Conclusion

In this paper, we propose a Reparameterized Residual Feature Network for lightweight image super-resolution. A multi branch structure is designed to capture the features of various patterns as much as possible and fuse them, then, reparameterization is introduced to enable complex multi-branch structures to be applied to lightweight networks. In the process of network training, a loss function based on Fourier transform is designed, which converts the image from spatial domain to frequency domain to guide the model to learn frequency information. Experiments have shown that the proposed method achieves better balance in performance and efficiency compared to other networks.

### References


[3] Xiaohan Ding, Yuchen Guo, Guiguang Ding, and Jungen Han. Acnet: Strengthening the kernel skeletons for powerful cnn via asymmetric convolution blocks. In *Proceedings of
the IEEE/CVF international conference on computer vision, pages 1911–1920, 2019. 3, 5
[27] Kai Zhang, Shuhang Gu, Radu Timofte, Zheng Hui, Xiumei Wang, Xinbo Gao, Dongliang Xiong, Shuai Liu, Ruipeng

