Wide Feature Projection with Fast and Memory-Economic Attention for Efficient Image Super-Resolution

Minghao Fu[1]
minghaofu@std.uestc.edu.cn
Dongyang Zhang[1]
dyzhang@std.uestc.edu.cn
Min Lei[1,2]
minlei@std.uestc.edu.cn
Kun He[1,2]
hekun@std.uestc.edu.cn
Changyu Li[1]
changyulve@std.uestc.edu.cn
Jie Shao[1,2]
shaojie@uestc.edu.cn

[1] University of Electronic Science and Technology of China
Chengdu, China

[2] Sichuan Artificial Intelligence Research Institute
Yibin, China

Abstract

With the development of efficient Super-Resolution (SR), many CNN-based methods adopt re-parameterization techniques to accelerate inference while training a wider network. However, the wide feature maps often lead to difficulty in reaching convergence due to information redundancy. To expand network width with a positive effect on restoration quality, we propose a novel Wide Feature Projector (WFP) module to get more benefits from wider feature. Besides, previous attention structures were computationally complex and occupied much memory. To address this issue, we investigate the peak memory consumption of attention structures in order to design a Fast and Memory-Economic Attention (FMEA) module, which factorizes the element-wise attention map to speed up inference and minimize memory consumption. Consequently, we propose a novel efficient SR network, termed as Wide Feature Projection Network (WFPN), which achieves a compelling super-resolution performance, and consistently beats its competitors in terms of computation complexity and memory cost.

1 Introduction

Image Super-Resolution (SR) is a popular research topic in computer vision, which aims at recovering Low-Resolution (LR) images to High-Resolution (HR) correspondence. Since the pioneering work SRCNN [4] was proposed, deep learning based methods have made a breakthrough in image restoration, but the running speed and memory occupation are still

challenges in SR tasks. Most existing SR models tend to employ a highly complicated network structure that requires large amounts of memory and parameters, thereby hindering the model deployment in resource-limited environment.

How to improve model efficiency and minimize model size has attracted growing attention recently [18, 33]. Previous studies suggest that using recurrent neural network [19, 25] or cascading structure [2] is feasible to reduce parameters, but such techniques are faced with high computation complexity. Others attempt to construct a lightweight model with fewer FLOPs. However, the number of FLOPs is not an equivalent judgment of running speed [33]. Another way is to directly reduce model depth and width, as some tiny models [2, 24] have a fast running speed and a low memory cost on account of their tiny size. However, simple structures always cannot fit high-frequency information perfectly, and training such models is challenging to obtain qualified SR images. In this work, we explore how to train a prominent and efficient SR model with wide features and re-parameterization technique.

Wide features, indicating augmenting the number of processed channels, can considerably improve network performance for SR tasks [30]. Recent works [28, 34] adopt an expand-and-squeeze structure to enhance learning local texture information from wide features. However, as the network width grows, gains of model performance are limited and even suffer from an unstable convergence, such as overfitting or gradient vanishing/exploding. This is because with the number of processed channels rising, the degree of information redundancy increases which makes the neural network more difficult to aggregate multiplied information from wider features. To address this issue, we propose a Wide Feature Projector (WFP) module to learn wide features through projecting channel information and maintaining spatial information simultaneously. Overall speaking, WFP improves the learning ability of wide features and sufficiently expands the network width to get advantages of over-parameterization during training.

Recently, structural re-parameterization [5] was proposed to reduce the number of parameters without compromising network performance. Some existing SR methods [28, 34] have used re-parameterization to achieve superior inference speed. In this work, we integrate the re-parameterization technique into the proposed Wide Feature Projection Block (WFPB), which is a typical multi-branch structure consisting of two WFPs and one standard 3×3 convolution block. The design of WFPB enables us to train heavy network but obtain lightweight network with no sacrifice of performance.

Attention is an effective mechanism to fully excite the capability of SR networks. However, conventional attention structures occupy a lot of memory and computation resources due to the multi-branch structure and high-complexity computation. Therefore, we investigate the sharing mechanism in the attention structure while involving identity mapping [10], and explore the relation between the memory consumption of each individual branch and the total. Based on the above discoveries, we propose a Fast and Memory-Economic Attention (FMEA) module, which allows WFPB to concentrate on more significant pixels to improve model discriminability whereas keep a fast running speed with consuming low memory.

The main contributions of our work can be summarized as follows:

- We propose a Wide Feature Projector (WFP) module to train a wider network. Our experiments show that WFP could largely alleviate training difficulties in the larger expanding ratio than the common expand-and-squeeze structure.
- We explore the factors of memory variation in attention by analyzing the running memory of representative structures. Through our careful analysis, we introduce a Fast and Memory-Economic Attention (FMEA) module that costs little resources with much improved performance.

- Equipped with FMEA and re-parameterization strategy, we further design a Wide Feature Projection network (WFPN). With lower memory consumption and computation complexity, WFPN demonstrates a comparable performance to state-of-the-art lightweight SR methods.

2 Related Work

Efficient image super-resolution. Due to the dilemma of huge memory cost and limited computing resources on mobile devices, it is imperative to develop efficient SR for real-world usage. Kim *et al*. [15] proposed a Very Deep SR (VDSR) network via residual learning to reduce the number of parameters. Ahn *et al*. [2] proposed an efficient CAscading Residual Network (CARN) to reduce FLOPs. Hui *et al*. [13, 14] used InforMation Distillation Network (IMDN) to compress number of filters per layer. FSRCNN [7] obtained high acceleration from traditional SRCNN [6], benefitting from its plain structure. RFDN [21] utilized features in residual connection for more efficient feature extraction. RLFN [17] used residual local feature learning to simplify feature aggregation that optimizes model inference time.

Re-parameterization. Recently, re-parameterization technique has shown its practicability in simplifying complex models. Zagoruyko *et al*. [31] proposed DiracNets with weight parameterization for neural networks to no longer add explicit skip connection. Ding *et al*. [5] applied a structural re-parameterization in VGG network to improve training capability. In SR tasks, Zhang *et al*. [34] utilized the re-parameterization strategy to build an Edge-oriented Convolution Block (ECB) to replace the standard 3×3 convolution. Wang *et al*. [28] introduced batch normalization into SR re-parameterization methods. However, the above SR methods based on re-parameterization suffer from complicated structure and optimization issues.

Attention mechanism. Recent works [4, 8, 11, 27, 29, 35] aim at guiding the network to augment the weight of important signals and alleviate unnecessary ones. SE-Net [11] is the first work to enhance information extraction through channel attention. CBAM [29] computes attention information using pooling and convolution to generate attention maps, which can be integrated into CNN. Zhang *et al*. [35] proposed Residual Channel Attention Network (RCAN) by introducing the channel attention mechanism into a modified residual block for SR. However, most of these methods adopt complex structures and time-consuming operations, which are extremely detrimental model inference speed and not appropriate to resource-limited devices.

3 Our Approach

We first present the overall network architecture in Section 3.1. In Section 3.2, we introduce our Wide Feature Projection Block (WFPB) and how to re-parameterize this architecture into a standard 3×3 convolution. Finally, we investigate memory consumption of multi-branch structures in Section 3.3, which motivates us to design Fast and Memory-Economic Attention (FMEA) in Section 3.4.

3.1 Network architecture

Previous SR networks adopt complex topologies as backbones. For instance, multiple branches [20] and dense connections [26] can enrich the feature representation without introducing many FLOPs, but concatenation and concurrency lead to high memory consumption and sacrifice the parallelism degree. We thus use a sequential model structure to improve inference speed and reduce network bandwidth.

The architecture of the proposed WFPN is depicted in Figure 1. WFPN applies Attention Block (AB) at the both head and tail for the channel's expansion and recovery, and alternatively stacks Residual Attention Blocks (RABs) and activation functions as the network body. Overall, the inference of WFPN can be explained as follows:

$$F_0 = h_{AB}(I_{LR}), \tag{1}$$

$$F_n = h^n_{RAB}(h^{n-1}_{RAB}(...h^0_{RAB}(F_0)...)), \tag{2}$$

$$I_{SR} = h_{AB}(F_n) + h_{UP}(I_{LR}), \tag{3}$$

where Eq. (1) represents feature extraction, and h_{AB} stands for the first attention block as shown in Figure 1(b), I_{LR} is the input LR images and F_0 is the initial feature maps. Eq. (2) represents feature learning, and h^n_{RAB} is the n-th Residual Attention Block (RAB). Eq. (3) represents the HR image restoration process and h_{UP} is the upsampler. In this work, we set $n = 16$ to achieve a trade-off between performance and efficiency. Specifically, we select the bilinear interpolation as a long-skip connection, and perform the upsampling operation at the end of network with a pixel-shuffle layer. We apply PReLU [9] as the activation function to each basic component.

3.2 Wide feature projection block

3.2.1 Wide feature projector

Wide Feature Projector (WFP) is proposed to learn a wide feature in a further way. As shown in Figure 1(d), we employ a $D \times C \times 1 \times 1$ convolution filter as a feature expander, and a $C \times D \times 3 \times 3$ filter as a feature squeezer, where C and D denote the channel numbers before and after feature squeezing. We adopt a 1×1 convolution as the feature projector paralleled with the 3×3 convolution, and insert a batch normalization layer in the middle for adding non-linearity property and accelerating convergence speed. In our experiments, we maximize the network width with the setting of expanding ratio $D/C = 4$ to yield the best performance.

This design is motivated by empirical phenomenons about wide SR network training. Some SR methods with re-parameterization [28, 34] employ an expand-and-squeeze structure as a basic component of sequential network to sense the information from its former block. They both set a small expanding ratio since their experiment results show that performance confronts a degradation when this hyperparameter continues rising. Besides, we found that if we train network with a larger expanding ratio, gradient vanishing/exploding may occur during training time, and even network cannot reach convergence due to overfitting. Such phenomenons indicate that as the network width increases, learning from wide features becomes very difficult, which triggers our interest: *since expanding network width with the re-parameterization is free, can we get more benefit from over-parameterization by further expanding wide features?* Intuitively we could use an identity mapping from residual connection [10] to reduce learning difficulty, but between the expanded and original features,

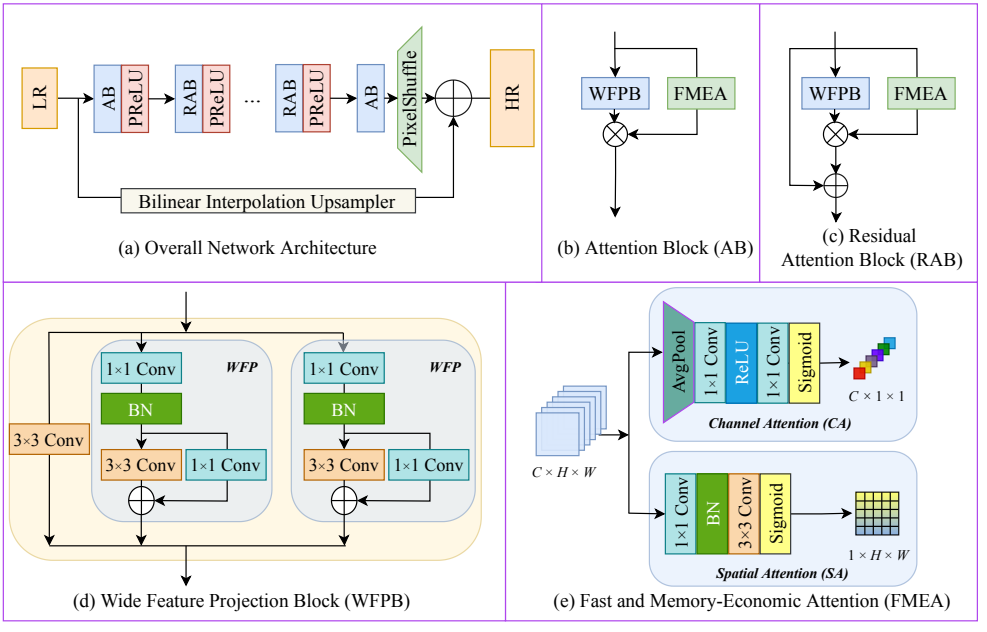

Figure 1: Network architecture of the proposed Wide Feature Projection Network (WFPN). $\bigoplus$ denotes element-wise summation, and $\bigotimes$ denotes element-wise multiplication.

the input and output are not of the same dimension, and thus we propose using a channel-wise compression as a bridge. We found that in different linear operations, a standard 1×1 convolution is the most appropriate for projecting wide features into narrow features. Thus, we select a $C \times D \times 1 \times 1$ convolution as a projector to preserve spatial information of wide features into the subsequent output. In the end, the projector conducts an element-wise summation with the 3×3 squeezer convolution. The output summation of squeezer and projector is calculated as follows:

$$F_j = \sum_{i}^{4n} (W_i \circledast f_i + k_i \times f_i),\ i = 1,2,...,4n,\ j = 1,2,...,n, \tag{4}$$

where f_i denotes the i-th input feature map and F_j corresponds the j-th output feature map. W_i represents the weight matrix of 3×3 convolution, and k_i represents the coefficient of channel-wise compression. $\sum k_i \times f_i$ serves as a projector for mapping information of wide features. The combination of projector and squeezer is explained as *spatial residual learning*: $k_i \times f_i$ learns the channel information and reserves the relative spatial information to the output, which could largely reduce the difficulties of training $W_i \circledast f_i$.

Based on WFP, Wide Feature Projection Block (WFPB) is devised as shown in Figure 1(d). Multi-branch WFP is regarded as feature extractor from wider features. To reduce training time and obtain the best performance, we set the branch number as 2. The additional 3×3 convolution could better propagate detailed information from the input feature maps. Although WFPB seems complicated, thanks to re-parameterization three branches in WFPB can be merged into a standard 3×3 convolution during inference. A batch normalization

(BN) layer is also used in each WFP, which is essential to introduce non-linearity and conducive to final results. Additionally, the entire WFPB is accessible to the re-parameterization technique, which is necessary for an efficient inference.

3.2.2 Re-parameterization in inference

The nice things about convolution is its associativity and linearity, which mean that continuous and parallel convolutions could be merged into one, meanwhile keep acceptive to other linear operations. Thus, we could improve the performance of standard convolution by training it in a more complicated formula, and all we need to do is re-parameterizing it in inference stage.

We now describe how to re-parameterize WFPB into a standard 3×3 convolution. After re-parameterization, output feature F can be calculated by the final weight and bias $\{K_{rep}, B_{rep}\}$ of convolution:

$$F = K_{rep} \circledast X + B_{rep}, \tag{5}$$

where X denotes the input feature. $\{K_{rep}, B_{rep}\}$ are calculated in inference as the left of Eq. (6):

$$\begin{cases} K_{rep} = K_{WFP_{(1)}} + K_{WFP_{(2)}} + K_n \\ B_{rep} = B_{WFP_{(1)}} + B_{WFP_{(2)}} + B_n \end{cases}, \quad \begin{cases} K_{WFP} = perm(K_e) \circledast (K_s + K_p \ pad \ 0) \\ B_{WFP} = (K_s + K_p \ pad \ 0) \circledast (B_e \ pad \ B_e) + B_s + B_p \end{cases} \tag{6}$$

The right of Eq. (6) explains how to represent WFP by merging different convolution kernels. $\{K_e, B_e\}$, $\{K_s, B_s\}$ and $\{K_p, B_p\}$ denote the weight and bias of 1×1 expander, 3×3 squeezer and 1×1 projector convolutions. Batch normalization has been folded in expander. Particularly, *perm* operation exchanges the first and second dimensions of the filter, and *pad* is a padding operation to adjust 1×1 kernel to 3×3 kernel.

3.3 Memory consumption analysis

It is suggested in [8] that network memory is mainly affected by four components: input feature memory M_{input}, output feature memory M_{output}, kept feature memory M_{kept} which is reserved for future usage, and parameter memory M_{net}. Because M_{input} and M_{output} are regular consumption in sequential models, and M_{net} is extremely small that can be ignored, we concentrate on the analysis of M_{kept}. To directly display memory consumption rather than analyzing it from experiences, we construct models with typical attention structures and identity mapping. By using the network shown in Figure 1(a) as the backbone and standard 3×3 convolution as components, we add attention layer or identity mapping as other branches to the basic components for observing different memory values in the stage of inference. The peak memory consumption of representative structures are summarized in Table 1.

We have the following observations from Table 1: **1)** When we add identity mapping $+x$ to $Conv(x)$, the peak memory consumption has a regular increment, because identity feature has to stay in memory until element-wise summation, which leads to an increment of M_{kept}. **2)** $CA(x)$ and $SA(x)$ have almost equivalent memory cost with $+x$. In the inference stage the model needs to generate an attention map for element-wise computation after the convolution block, and x has to be kept in memory to conduct such computation while $Conv(x)$ is

Identity		Attention		Identity + Attention	
Base Block	Memory (M)	Base Block	Memory (M)	Base Block	Memory (M)
$Conv(x)$	39.92	$Conv(x) \times CA(x)$	54.27	$Conv(x) \times CA(x) + x$	54.27
$Conv(x) + x$	54.26	$Conv(x) \times SA(x)$	54.49	$Conv(x) \times SA(x) + x$	54.49
$Conv(x) + 2x$	69.00	$Conv(x) \times PA(x)$	69.00	$Conv(x) \times PA(x) + x$	69.00

Table 1: Comparison of peak memory consumptions of different network structures. All models are validated on the task of $\times 4$ upscaling the single image to 1280×720 resolution. x represents input data, $Conv(x)$ denotes convolution layer, $+x$ indicates the identity mapping, and $+2x$ is the double identity mappings. CA, SA, PA represent Channel Attention [11], Spatial Attention [29], and Pixel Attention [36], respectively.

processing. **3)** The above two observations indicate that attention layer and identity mapping both increase peak memory consumption, but $CA(x) + x$ or $SA(x) + x$ have almost the same memory cost as $+x$. This observation can be interpreted as *identity memory sharing*: if the multi-branch structure has an identity mapping, other branch can utilize identity feature to generate a target output conditioned on it, rather than storing data twice. **4)** $CA(x)$ and $SA(x)$ have tiny divergence memory cost with that of $+x$, and $PA(x)$ has a memory expense much larger than them, nearly approaching that of $+2x$. Although identity mapping could share memory with attention layer, extra memory is still produced by the generated attention map. For channel attention and spatial attention, their attention map sizes are $C \times 1 \times 1$ and $1 \times H \times W$, which are much smaller than input data $C \times H \times W$. However, the attention map of pixel attention has the equivalent size with output data for an element-wise multiplication. Therefore, a valid method to reduce peak memory consumption is to avoid generating a large attention map.

3.4 Fast and memory-economic attention

Overview of our FMEA is represented in Figure 1(e). It consists of a spatial attention branch and a channel attention branch, which are broadly utilized in SR network. From Section 3.3, it is clear that there is much more memory spent on pixel attention than channel attention and spatial attention. Thus, FMEA splits a 3-dimensional element-wise attention into a 1-dimensional sequence and a 2-dimensional matrix. Therefore, FMEA M_{kept} has a size of $C \times 1 \times 1 + 1 \times H \times W$, which is negligible compared with $C \times H \times W$, the size of a element-wise attention map has to reserve in memory.

In order to reduce the computation burden, we generate attention maps with low-computation operations. Unlike CBAM [29], FMEA avoids concatenation and multi-step pooling to ensure fast running speed. For spatial attention, we take up a 1×1 convolution to aggregate channels into a single feature map, a batch normalization layer inside to speed up convergence, and a 3×3 convolution to enlarge the receptive field. Using a 1×1 convolution in advance to reduce dimensionality is an effective manipulation for that squeezing channels could help reduce memory and computation in square ratio. For channel attention, we adopt the same squeeze-and-excitation structure as [11]. The bottleneck structure could largely reduce parameter filter size which results in a little amount of computation, and such an attention method is proven to achieve superior effectiveness in previous networks [11, 35].

Dataset	Scale	Bicubic	CARN [0] (1592K)	IMDN [14] (715K)	RFDN [21] (550K)	RLFN [17] (543K)	WFPN (ours) (633K)
Set5	×2	33.66 / 0.9299	37.76 / 0.9590	38.00 / 0.9605	38.05 / 0.9606	38.07 / 0.9607	38.08 / 0.9607
	×4	28.42 / 0.8104	32.13 / 0.8937	32.21 / 0.8948	32.24 / 0.8952	32.24 / 0.8952	32.25 / 0.8954
Set14	×2	30.24 / 0.8688	33.52 / 0.9166	33.63 / 0.9177	33.68 / 0.9184	33.72 / 0.9187	33.69 / 0.9179
	×4	26.00 / 0.7027	28.60 / 0.7806	28.58 / 0.7811	28.61 / 0.7819	28.62 / 0.7813	28.65 / 0.7813
B100	×2	29.56 / 0.8431	32.09 / 0.8978	32.19 / 0.8996	32.16 / 0.8994	32.22 / 0.9000	32.24 / 0.9002
	×4	25.96 / 0.6675	27.58 / 0.7349	27.56 / 0.7353	27.57 / 0.7360	27.60 / 0.7364	27.62 / 0.7367
Urban100	×2	26.88 / 0.8403	31.92 / 0.9256	32.17 / 0.9283	32.12 / 0.9278	32.33 / 0.9299	32.29 / 0.9285
	×4	23.14 / 0.6577	26.07 / 0.7837	26.04 / 0.7838	26.11 / 0.7858	26.17 / 0.7877	26.19 / 0.7878
Manga109	×2	31.01 / 0.8923	38.36 / 0.9765	38.88 / 0.9774	38.88 / 0.9773	–	38.90 / 0.9773
	×4	26.66 / 0.7512	30.47 / 0.9084	30.45 / 0.9075	30.58 / 0.9089	–	30.55 / 0.9085

Table 2: Performance comparison on benchmark datasets. Number of model parameters is computed on ×4 task. Red indicates the best and blue indicates the second best.

Method	Params (K)	FLOPs (G)	Activations (M)	Runtime (ms)	Memory (M)
IMDN	894	58.53	154.14	127.55	471.76
RLFN	543	33.99	112.03	100.46	421.26
FMEN	769	50.28	118.23	99.41	262.32
WFPN	632	40.73	76.81	110.23	245.49

Table 3: Efficiency comparison for ×4 upscaling, with PyTorch 1.4.0, CUDA Toolkit 10.0.130, on an NVIDIA Titan X GPU.

4 Experiments

4.1 Dataset and implementation details

We use the high-quality datasets DIV2K [1] and Flickr2K as the training set. The validation sets are Set5 [3], Set14 [32], BSD100 [22], Urban100 [12] and Manga109 [23]. Results are evaluated on the luminance channel of YCbCr space with PSNR and SSIM as metrics.

We randomly crop 96×96 pixels of an HR image as training input. We set mini-batch size to 32 as each training iteration input, and adopt data augmentation on the training set to improve training effect by random rotations and flipping. We train different upscaling models with the ADAM optimizer [16] and L1 loss function for 2000 epochs. The initial learning rate is set to 4×10^{-4}, and decreases half per 2×10^6 iterations for total 2×10^7 iterations. Training is conducted on an NVIDIA Titan X GPU.

4.2 Comparison with competitive methods

Performance comparison. We compare our WFPN with other lightweight SR models, including CARN [0], IMDN [14], RFDN [21], and RLFN [17]. Among them, RLFN and RFDN respectively won the first place of NTIRE 2022 efficient SR challenge [18] and AIM 2020 challenge on efficient SR [33]. The quantitative comparisons for ×2 and ×4 upscalings on benchmark datasets are shown in Table 2. We can find that WFPN shows more competitive effect, and achieves better performance in PSNR / SSIM than these methods. Visual comparisons are illustrated in Figure 2. WFPN yields more visually pleasant patterns in the selected Set14 and B100 images, compared with other methods.

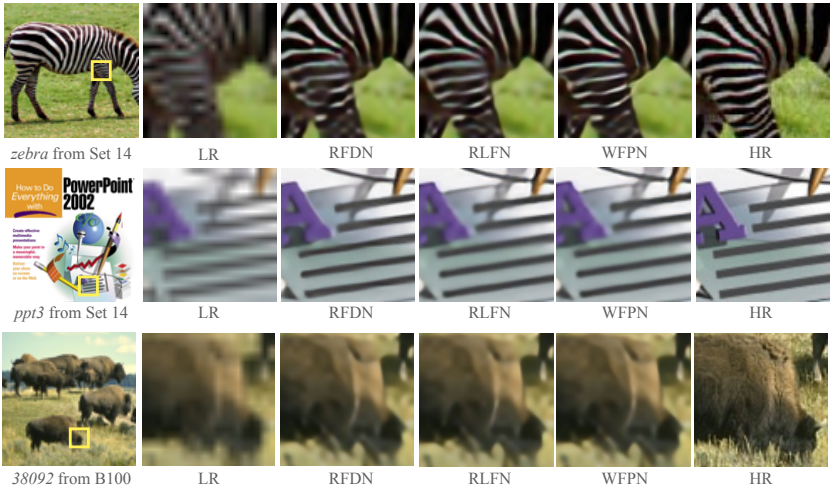

Figure 2: Visual results on Set14 and B100 for $\times 4$ upscaling.

Expanding Ratio	1	2	3	4	5
WFP	31.95	32.09	32.12	32.15	32.09
ESC	31.94	32.06	32.07	-	-

Table 4: PSNR results evaluated on Set5 of different expanding ratios of WFP and ESC. $-$ denotes there is a gradient vanishing/exploding problem or stopping converging untimely during training.

Efficiency comparison. We also evaluate resource and time-consuming metrics. We select three recent state-of-the-art lightweight SR methods, IMDN [14], RLFN [17] and FMEN [3] as comparison models in terms of efficiency. As shown in Table 3, WFPN achieves the lowest memory and activations, the second lowest FLOPs and parameters. The results indicate that our analysis about peak memory consumption has a good implication for real-world applications, and re-parameterization could largely reduce model parameters.

4.3 Ablation study

All the ablation experiments are conducted on the $\times 4$ model. Especially, we record the results in 4×10^6 iterations.

Expanding ratio. We adjust the expanding ratio of WFP from 1 to 5. Table 4 implies that setting $ratio = 4$ works best. According to trends of PSNR, with the expanding ratio increasing, WFP is improved gradually until the number of channels overloads the training capability.

WFP vs. expand-and-squeeze convolution. We replace WFP in our network with Expand-and-Squeeze Convolution (ESC) and then perform the identical adjustment of expanding ratio. ESC has the same expander and squeezer convolutions as WFP but no projector. From Table 4 we observe that with the expanding ratio increasing, ESC is confronted with a convergence problem during training. As a contrast, WFP can keep training stability no matter how ratio grows. These results verify that WFP performs better on training a wider network than ESC.

Structure	Set5	Set14
2×WFP + 3×3 conv	32.05	28.55
2×WFP + 1×1 conv	32.02	28.45
2×WFP + identity	31.98	28.43
2×WFP	29.56	26.93

Branches	Set5	Set14
1	32.01	28.54
2	32.05	28.55
3	32.02	28.57
4	31.95	28.53

Table 5: Ablation studies on WFPB structure.

Attention Type	Params (K)	Multi-Adds (G)	Memory (M)	Set5	Set14
WFPN	633	35.87	58.62	32.15	33.65
WFPN_SA	623	35.81	58.57	32.12	33.62
WFPN_CA	632	35.79	58.54	32.08	33.59
WFPN_Plain	621	35.73	58.48	32.02	33.42
WFPN_PA	691	39.69	73.51	32.16	33.61

Table 6: Effect of FMEA.

Overall structure of WFPB. As shown in Table 5, we explore different choices of WFPB structure and the most suitable number of WFP branches. **1)** Replacing the 3×3 convolution with a 1×1 convolution would not bring an apparent degradation. They both could serve as the base performance insurance of WFPB. **2)** Identity mapping is also a valid method, but it is still much worse than the convolution layer. **3)** Without the convolution or identity mapping, WFP cannot serve as a valueable branch to conduct feature learning. **4)** We change the number of WFP branches in WFPB. It is observed that two WFPs lead to the best performance, but the PSNR saturates if we further increase the branch number.

Effectiveness and efficiency of FMEA. To demonstrate the impact of the proposed FMEA module, we select our WFPN as the basic network, and keep only Spatial Attention (SA) branch of FMEA as WFPN_SA, Channel Attention (CA) branch of FMEA as WFPN_CA, remove FMEA as WFPN_Plain, and replace FMEA with Pixel Attention (PA) [36] as WFPN_PA. Table 4.3 shows that CA and SA of FMEA both improve our WFPN, while SA contributes more to the restoration quality. By comparing WFPN and WFPN_Plain, we found that FMEA only consumes little in parameters, computation resource, and memory. WFPN_PA has a large resource consumption rise compared with WFPN_Plain, which justifies our memory analysis about multi-branch structure in Section 3.3.

4.4 Conclusion

In this paper, we construct Wide Feature Projection Network (WFPN) for efficient super-resolution. To alleviate training difficulty in wide features, we propose a Wide Feature Projection Block (WFPB) based on Wide Feature Projection (WFP) to train a wider network, which is inference-efficient with the help of re-parameterization technique. Through investigating peak memory consumption of multi-branch network, we design a Fast and Memory-Economic Attention (FMEA) by factorizing attention map to minimize resources of attention mechanism. WFPN achieves better performance than previous methods with low resource consumption and a fast running speed.

Acknowledgments This work was supported by the National Natural Science Foundation of China (grant No. 61832001) and the Ministry of Science and Technology of China (grant No. G2022036009L).

References

[1] Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1122–1131, 2017.

[2] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part X*, pages 256–272, 2018.

[3] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie-Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *British Machine Vision Conference, BMVC 2012, Surrey, UK, September 3-7, 2012*, pages 1–10, 2012.

[4] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 11065–11074, 2019.

[5] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 13733–13742, 2021.

[6] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(2): 295–307, 2016.

[7] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II*, pages 391–407, 2016.

[8] Zongcai Du, Ding Liu, Jie Liu, Jie Tang, Gangshan Wu, and Lean Fu. Fast and memory-efficient network towards efficient image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2022, New Orleans, LA, USA, June 19-24, 2022*, pages 853–862, 2022.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 1026–1034, 2015.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778, 2016.

[11] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 7132–7141, 2018.

[12] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 5197–5206, 2015.

[13] Zheng Hui, Xiumei Wang, and Xinbo Gao. Fast and accurate single image super-resolution via information distillation network. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 723–731, 2018.

[14] Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang. Lightweight image super-resolution with information multi-distillation network. In *Proceedings of the 27th ACM International Conference on Multimedia, MM 2019, Nice, France, October 21-25, 2019*, pages 2024–2032, 2019.

[15] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 1646–1654, 2016.

[16] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[17] Fangyuan Kong, Mingxi Li, Songwei Liu, Ding Liu, Jingwen He, Yang Bai, Fangmin Chen, and Lean Fu. Residual local feature network for efficient super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2022, New Orleans, LA, USA, June 19-24, 2022*, pages 766–776, 2022.

[18] Yawei Li, Kai Zhang, Radu Timofte, Luc Van Gool, Fangyuan Kong, Mingxi Li, Songwei Liu, Zongcai Du, Ding Liu, Chenhui Zhou, Jingyi Chen, Qingrui Han, Zheyuan Li, Yingqi Liu, Xiangyu Chen, Haoming Cai, Yu Qiao, Chao Dong, Long Sun, Jinshan Pan, Yi Zhu, Zhikai Zong, Xiaoxiao Liu, Zheng Hui, Tao Yang, Peiran Ren, Xuansong Xie, Xian-Sheng Hua, Yanbo Wang, Xiaozhong Ji, Chuming Lin, Donghao Luo, Ying Tai, Chengjie Wang, Zhizhong Zhang, Yuan Xie, Shen Cheng, Ziwei Luo, Lei Yu, Zhihong Wen, Qi Wu, Youwei Li, Haoqiang Fan, Jian Sun, Shuaicheng Liu, Yuanfei Huang, Meiguang Jin, Hua Huang, Jing Liu, Xinjian Zhang, Yan Wang, Lingshun Long, Gen Li, Yuanfan Zhang, Zuowei Cao, Lei Sun, Panaetov Alexander, Yucong Wang, Minjie Cai, Li Wang, Lu Tian, Zheyuan Wang, Hongbing Ma, Jie Liu, Chao Chen, Yidong Cai, and et al. NTIRE 2022 challenge on efficient super-resolution: Methods and results. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2022, New Orleans, LA, USA, June 19-24, 2022*, pages 1062–1102, 2022.

[19] Zhen Li, Jinglei Yang, Zheng Liu, Xiaomin Yang, Gwanggil Jeon, and Wei Wu. Feedback network for image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 3867–3876, 2019.

[20] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1132–1140, 2017.

[21] Jie Liu, Jie Tang, and Gangshan Wu. Residual feature distillation network for lightweight image super-resolution. In *Computer Vision - ECCV 2020 Workshops - Glasgow, UK, August 23-28, 2020, Proceedings, Part III*, pages 41–55, 2020.

[22] David R. Martin, Charless C. Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the Eighth International Conference On Computer Vision (ICCV-01), Vancouver, British Columbia, Canada, July 7-14, 2001 - Volume 2*, pages 416–425, 2001.

[23] Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. Sketch-based manga retrieval using manga109 dataset. *Multim. Tools Appl.*, 76(20):21811–21838, 2017.

[24] Wenzhe Shi, Jose Caballero, Ferenc Huszar, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 1874–1883, 2016.

[25] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2790–2798, 2017.

[26] Tong Tong, Gen Li, Xiejie Liu, and Qinquan Gao. Image super-resolution using dense skip connections. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 4809–4817, 2017.

[27] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 11531–11539, 2020.

[28] Xintao Wang, Chao Dong, and Ying Shan. Repsr: Training efficient vgg-style super-resolution networks with structural re-parameterization and batch normalization. *CoRR*, abs/2205.05671, 2022.

[29] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. CBAM: convolutional block attention module. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII*, pages 3–19, 2018.

[30] Jiahui Yu, Yuchen Fan, and Thomas S. Huang. Wide activation for efficient image and video super-resolution. In *30th British Machine Vision Conference 2019, BMVC 2019, Cardiff, UK, September 9-12, 2019*, page 189, 2019.

[31] Sergey Zagoruyko and Nikos Komodakis. Diracnets: Training very deep neural networks without skip-connections. *CoRR*, abs/1706.00388, 2017.

[32] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces - 7th International Conference, Avignon, France, June 24-30, 2010, Revised Selected Papers*, pages 711–730, 2010.

[33] Kai Zhang, Martin Danelljan, Yawei Li, Radu Timofte, Jie Liu, Jie Tang, Gangshan Wu, Yu Zhu, Xiangyu He, Wenjie Xu, Chenghua Li, Cong Leng, Jian Cheng, Guangyang Wu, Wenyi Wang, Xiaohong Liu, Hengyuan Zhao, Xiangtao Kong, Jingwen He, Yu Qiao, Chao Dong, Xiaotong Luo, Liang Chen, Jiangtao Zhang, Maitreya Suin, Kuldeep Purohit, A. N. Rajagopalan, Xiaochuan Li, Zhiqiang Lang, Jiangtao Nie, Wei Wei, Lei Zhang, Abdul Muqeet, Jiwon Hwang, Subin Yang, Jung Heum Kang, Sung-Ho Bae, Yongwoo Kim, Yanyun Qu, Geun-Woo Jeon, Jun-Ho Choi, Jun-Hyuk Kim, Jong-Seok Lee, Steven Marty, Éric Marty, Dongliang Xiong, Siang Chen, Lin Zha, Jiande Jiang, Xinbo Gao, Wen Lu, Haicheng Wang, Vineeth Bhaskara, Alex Levinshtein, Stavros Tsogkas, Allan D. Jepson, Xiangzhen Kong, Tongtong Zhao, Shanshan Zhao, Hrishikesh P. S, Densen Puthussery, C. V. Jiji, Nan Nan, Shuai Liu, Jie Cai, Zibo Meng, Jiaming Ding, Chiu Man Ho, Xuehui Wang, Qiong Yan, Yuzhi Zhao, Long Chen, Long Sun, Wenhao Wang, Zhenbing Liu, Rushi Lan, Rao Muhammad Umer, and Christian Micheloni. AIM 2020 challenge on efficient super-resolution: Methods and results. In *Computer Vision - ECCV 2020 Workshops - Glasgow, UK, August 23-28, 2020, Proceedings, Part III*, pages 5–40, 2020.

[34] Xindong Zhang, Hui Zeng, and Lei Zhang. Edge-oriented convolution block for real-time super resolution on mobile devices. In *MM '21: ACM Multimedia Conference, Virtual Event, China, October 20 - 24, 2021*, pages 4034–4043, 2021.

[35] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII*, pages 294–310, 2018.

[36] Hengyuan Zhao, Xiangtao Kong, Jingwen He, Yu Qiao, and Chao Dong. Efficient image super-resolution using pixel attention. In *Computer Vision - ECCV 2020 Workshops - Glasgow, UK, August 23-28, 2020, Proceedings, Part III*, pages 56–72, 2020.