Computing and Informatics, Vol. 43, 2024, 414–437, doi: 10.31577/cai_2024_2_414

BTAN: LIGHTWEIGHT SUPER-RESOLUTION NETWORK WITH TARGET TRANSFORM AND ATTENTION

Pan Wang, Zedong Wu, Zicheng Ding, Bochuan Zheng*

School of Computer Science
China West Normal University
Nanchong, 637009, China
e-mail: {1573049371, 1780707273, 631751336}@qq.com,
zhengbc@vip.163.com

Abstract. In the realm of single-image super-resolution (SISR), generating high-resolution (HR) images from a low-resolution (LR) input remains a challenging task. While deep neural networks have shown promising results, they often require significant computational resources. To address this issue, we introduce a lightweight convolutional neural network, named BTAN, that leverages the connection between LR and HR images to enhance performance without increasing the number of parameters. Our approach includes a target transform module that adjusts output features to match the target distribution and improve reconstruction quality, as well as a spatial and channel-wise attention module that modulates feature maps based on visual attention at multiple layers. We demonstrate the effectiveness of our approach on four benchmark datasets, showcasing superior accuracy, efficiency, and visual quality when compared to state-of-the-art methods.

Keywords: Image super-resolution, light-weight network, target transform, attention mechanism, deep learning

Mathematics Subject Classification 2010: 68U10

* Corresponding author

1 INTRODUCTION

Single image super-resolution (SISR) is a notoriously challenging problem in low-level computer vision tasks because the high-resolution (HR) space is mismatched with the low-resolution (LR) space [1]. Every LR image can potentially match many HR image patches, leading to the problem of successfully restoring. Until recently, many convolutional neural networks (CNN) based methods have been proposed [2, 3, 4, 5, 6, 7, 8, 9], which provide an outstanding performance than the conventional methods. For example, compared with the interpolated methods, the recently transformer-based methods [10, 11, 12] show superior advantages in upsampling accuracy. However, most accurate models are enormous in size and heavy in processing time, which cannot meet the requirement of real-world applications. To solve the SISR problem in a practical way, we need to consider both model accuracy and model efficiency.

Deep learning makes big help to the model accuracy, SRCNN [13] using only three convolutional layers has successfully improved the performance of SISR a lot. Since then with the success of deep learning in many other computer-vision tasks, many methods have been applied to SISR area, such as residual connection [14, 15], hourglass structure [16], recursive learning [17, 18], dense connection [19, 20], attention mechanism [21], and transformer [10, 11, 12]. These methods which have been proven in other computer-vision tasks can also be adopted in SISR. However, simply applying these advanced technologies does not always promise better performance. Stacking a deeper network can somehow receive an improvement in performance, but a giant model immediately leads to the problem of hard training and time expenditure. It seems that using more residual blocks can get a nice result; however, there is still a limitation to putting that to an extremity. To address the problem of gradient vanishing, adopting dense connection can enrich the pathways of the network which finally help with the results. Channel attention and space attention better utilize the extra channel and space information to help with upsampling from an information-deficient LR image. All the above methods can contribute to the model's accuracy, but they mostly need to add some blocks or adopt some mechanism that will add an extra burden on the original model. Can we change the network's inner running state without the expense of adding more parameters?

Global residual learning has been proven to improve VDSR significantly [14]; it forms a pathway to let input directly flow to the output and let the main network learn the residual part of the LR and HR. The network backbone learning target changes from HR to the residual of HR and LR. The network output takes the sum of the backbone target and input, and we know that the output will be close to HR through training. So we can formally define a way to reversely calculate the network backbone target. Let $\mathbf{T}$ be the network backbone target, and when there is no modification at the bottom of the network, the raw backbone target T_{raw} approximately equal to HR output, as

$$T_{raw} \approx I_{HR}, \tag{1}$$

when input flow as a residual to the backbone target, the sum of both is approximately equal to HR, so the residual backbone target $T_{residual}$ can be expressed as

$$I_{LR} + T_{residual} \approx I_{HR}, \tag{2}$$

$$T_{residual} \approx I_{HR} - I_{LR}, \tag{3}$$

where I_{LR} denotes the LR input, I_{HR} denotes the HR ground truth.

Changing the network backbone learning target will not cost any extra parameters, and even, if found, a better target will lessen the burden of the network and the network can have a better result. Kong and Fowlkes [22] proposed using the predictive filter to address the problem of SR which generates a good result. In their model, the network backbone learning target learns the quotient of HR and LR and has a generally better result. Here the backbone target elementwise multiplies the LR input, and finally approximately equal to the HR ground truth, expressed as

$$I_{LR} \odot T_{quotient} \approx I_{HR}, \tag{4}$$

$$T_{quotient} \approx I_{HR}/I_{LR}. \tag{5}$$

By observing these facts, we wonder if there may be a better relationship between HR and LR to learn in backbone target. The above relationships between LR and backbone target are depicted in Figure 1.

By exploring the above phenomena, we propose a network backbone target transform attention dense network (BTAN) for lightweight super-resolution. We first build a lean base network on a modified DenseNet [23]. The enriched pathways in DenseNet help with ease of training and bring superior performance. Drawing inspiration from the demonstrated efficacy of the Channel-Spatial Attention Mechanism (CSAM) in enhancing feature representation and image reconstruction quality [24], we have integrated this mechanism into our basic network architecture. This strategic inclusion aims to synergistically leverage both channel-wise and spatial information, significantly enhancing the network's ability to discern and emphasize pivotal features for improved super-resolution performance. And at the end of our network, we experiment with different network targets which are different algebra equations of HR and LR. Finally, we find a better network backbone target that has the best performance on our base network. By changing the backbone target, we realize the aim of improving model accuracy without adding extra parameters. In addition to the BTAN model, we also discussed the effectiveness of using different patch sizes in training an SR model, we found at a limitation, a larger patch size will guarantee a better training result.

In summary, our main contributions are as follows:

- We propose a novel convolutional neural network called BTAN that incorporates target transform and attention mechanism for image super-resolution.
- We design a target transform module that learns to adjust the output features to match the target distribution and improve the reconstruction quality.

- We introduce a spatial and channel-wise attention module that dynamically modulates the feature maps according to the visual attention at multiple layers.
- We evaluate our method on four benchmark datasets and show that it outperforms state-of-the-art methods in terms of accuracy, efficiency, and visual quality.

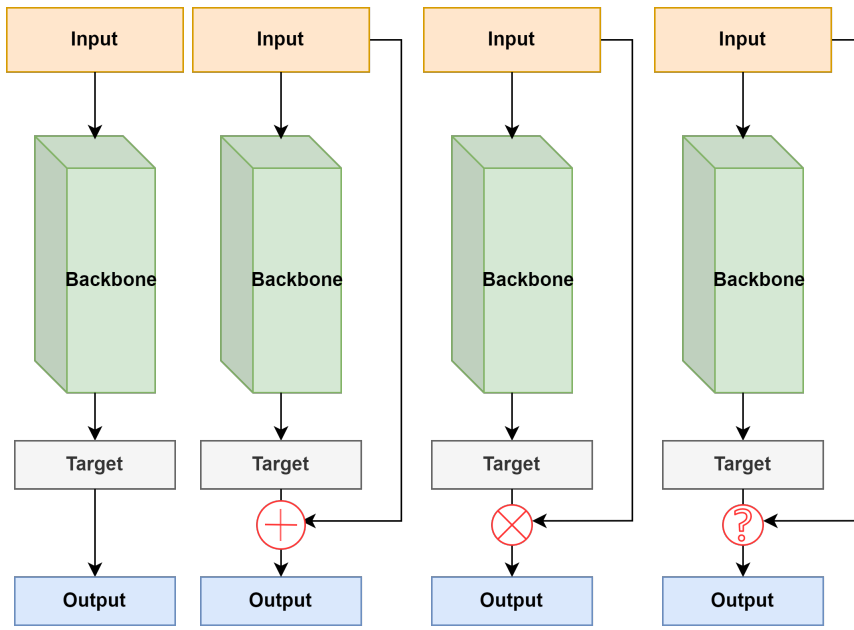


Figure 1. Different network backbone target

2 RELATED WORK

Recently, the deep learning techniques applied to Single Image Super Resolution (SISR) make this area rapidly progress. The SISR focuses on restoring high-quality images from low-resolution images, the prominent aim is to accurately learn the mapping between these two. So we will first review the accurate deep learning-based SISR in Section 2.1. Another requirement of SISR is efficiency, a good model which has small parameters and calculations can better suit real-world applications. We will review it in Section 2.2.

2.1 Accurate Image Super Resolution

Super-resolution (SR) aims to generate a high-resolution (HR) image from a low-resolution (LR) input image. Accurate image super-resolution is crucial for a wide range of computer vision applications such as medical imaging, satellite imagery, and surveillance. In recent years, several deep learning-based methods have been proposed to improve the accuracy of super-resolution.

One of the earliest deep learning-based methods is SRCNN proposed by Dong et al. [13] which utilizes a three-layer convolutional neural network (CNN) to learn an end-to-end mapping from LR to HR images. Later, Kim et al. [14] introduced a deeper model, VDSR, that includes 20 convolutional layers and residual connections to improve the accuracy of super-resolution.

One popular approach is the use of deep neural networks, such as the Residual Dense Network (RDN) proposed by Zhang et al. [21]. RDN is a deep neural network that uses densely connected residual blocks to extract and integrate multi-scale features.

To further improve the accuracy, SRGAN proposed by Ledig et al. utilizes a generative adversarial network (GAN) [25] to generate realistic and sharp HR [26]. Similarly, ESRGAN proposed by Wang et al. enhances the SRGAN model by incorporating a residual-in-residual dense block and a perceptual loss function to produce even more accurate super-resolved images [27].

Another approach to improve the accuracy of super-resolution is by incorporating attention mechanisms. For instance, RCAN proposed by Zhang et al. utilizes a residual channel attention network to selectively focus on the most informative image features during the SR process, resulting in a significant improvement in accuracy [20].

In summary, accurate image super-resolution has been greatly improved by deep learning-based methods that incorporate various techniques such as residual connections, GANs, and attention mechanisms. Although these methods have achieved state-of-the-art performance in terms of objective and subjective image quality metrics, they unavoidably introduce new structures at the expense of more new parameters.

2.2 Efficient Image Super Resolution

Lightweight image super-resolution is an important research area in computer vision, where the goal is to develop computationally efficient and parameter-reduced methods that can generate high-quality high-resolution images from low-resolution images. In this related work, we will discuss some of the recent developments in lightweight image super-resolution methods.

Building a lean model for efficient image super-resolution can have three main pathways, first is building a normal good model and then compressing the model using model distil or model pruning, second is using the technique of recursive learning, third is directly building a lean and impressive model.

To address the computational cost issue, Shi et al. proposed an efficient sub-pixel convolutional neural network (ESPCN) for image super-resolution [28]. The ESPCN method utilizes sub-pixel convolutional layers to upsample low-resolution feature maps, thus reducing the computational cost of the method while maintaining high accuracy.

Another approach to lightweight image super-resolution is through the use of deeper neural network architectures. Kim et al. proposed the Deeply-Recursive Convolutional Network (DRCN) for image super-resolution [17]. DRCN employs a deep neural network architecture with recursive layers to learn the mapping between low-resolution and high-resolution images. This method achieves the state-of-the-art performance with a relatively small number of parameters.

Memory networks (MemNet) are another popular approach to the lightweight image super-resolution [29]. MemNet employs a memory block architecture that can learn complex mappings between low-resolution and high-resolution images while maintaining a small number of parameters. This method achieves the state-of-the-art performance on benchmark datasets with reduced computational cost and the number of parameters.

Recently, several methods have been proposed that employ residual connections to improve the accuracy of super-resolution. For example, the Deep Recursive Residual Network (DRRN) [18] employs a deep neural network architecture with residual connections to learn the mapping between low-resolution and high-resolution images. The method achieves state-of-the-art performance with a relatively small number of parameters.

Some of these methods simplify the network structure by reducing the number of layers or channels [30, 31], while others introduce novel modules or operations to enhance the feature extraction or reconstruction ability [32, 33]. However, most of these methods ignore the close relationship between LR and HR images, which can be exploited to improve the SR quality without increasing the model complexity.

In conclusion, lightweight image super-resolution is an important research area in computer vision. Recent developments in this field have focused on improving the accuracy of super-resolution while reducing the computational cost and number of parameters. These methods have employed sub-pixel convolutional layers, deeper neural network architectures, memory block architectures, residual connections, content-aware residual blocks, and attention mechanisms to achieve state-of-the-art performance with reduced computational cost and the number of parameters.

3 PROPOSED METHOD

3.1 Network Structure

As mentioned in Section 1, we propose one lightweight super resolution model, BTAN, as shown in Figure 2, which mainly consisted of three parts, feature extraction (FE), Channel-Spatial Attention Dense Block Group (CADBG), and backbone

output Transform (Trans). Our BTAN model is based on RDN [20]. We refined the basic block of RDN with Channel-Spatial Attention Module (CSAM) as our new basic block, CADB is shown in Figure 2 b). We stack a few of these CADB blocks into a group to form our network backbone. Except for the backbone, we use nearest upsampling to upsample our input, forming an algebra transformation between backbone output and upsampled input.

Different from most other light-weight models, we choose to build our model from a concise yet fine baseline model. We start from a very small model, this will give us three folds of benefits, firstly it will simplify the model intro-relationship and allow us to better explore the correlations of different parts, secondly it will save the training resources including time resource and equipment resource, thirdly by exploring all the other none-parameter-increasing techniques, if we can improve the model performance, that will exactly echo with our proposed light-weight aim – improving model accuracy without adding an extra parameter.

To explicitly express the network flow in BTAN, let I_{LR} be the input and H be a convolution function. Then we can define the extracted feature F_{FE} as

$$F_{FE} = H_{FE}(I_{LR}). \tag{6}$$

And then, the extracted feature will flow into a group of CADBs, the feature will pass through each CADB and then at the end concatenate all the previous features and fuse into one share. Let F^i as the i^{th} CADB output feature, H^i_{CADB} as the i^{th} CADB convolve function, then we have:

$$\begin{aligned} F_i &= H^i_{CADB}(F_{i-1}) \\ &= H^i_{CADB}\left(H^{i-1}_{CADB}\left(\dots\left(H^1_{CADB}(F_1)\right)\dots\right)\right), \end{aligned} \tag{7}$$

finally, we let H_{GFF} as the concatenate fuse function, expressed as

$$F_{CADBG} = H_{GFF}([F_1, F_2, \dots, F_i]), \tag{8}$$

where F_{CADBG} denotes the output feature passed through CADBG block. Here H_{GFF} is a pointwise convolution operation which can shrink the concatenated channels into a normal number of channels. The global feature fusion will try to capture each level of information, to form a more integrated global feature.

After extracting the global and local features from the LR space, we upsample the backbone output to the same size of output with sub-pixel upsampling. And then also directly upsample the raw input to the same size with a simple nearest upsampling. Combined these two features, the backbone output feature and LR feature, with some generic relationship, we formed the transformed output feature F_{trans}, expressed as

$$F_{trans} = H_{nearUP}(I_{LR}) * H_{subpixUP}(F_{CADBG}), \tag{9}$$

where H_{nearUP} denotes the nearest upsampling operation, $H_{subpixUP}$ denotes the subpixel [28] upsampling operation. The $*$ operation represents a form of a functional relationship between these two upsampled features. And this is what we will explore in our paper, to see the details of final chosen $*$ operation, please refer to Table 2.

And finally, with a last simple 3×3 convolution operation, we get our model output I_{SR}, expressed as

$$\begin{aligned} I_{SR} &= H_{lastconv}(F_{trans}) \\ &= H_{BATN}(I_{LR}), \end{aligned} \tag{10}$$

where $H_{lastconv}$ denotes the last convolution operation, H_{BATN} denotes the function of our BATN.

3.2 Channel-Spatial Attention Dense Block

Now we present details about our proposed Channel-Spatial Attention Dense Block (CADB) in Figure 2 b). It mainly consists of three parts, dense connection, local feature fusion (LFF) and channel-spatial attention mechanism (CSAM). The introduced dense connection can help fully extract the features that pass through the network and also lessen the training difficulty. To maintain a consistent feature channel, use a local feature fusion to reduce the enlarged feature channels caused by dense connections. While the Local Feature Fusion (LFF) effectively amalgamates the diverse features extracted, the CSAM elevates this process by offering more adaptive and nuanced selection. It recognizes and emphasizes the distinct characteristics inherent in different channels and spatial blocks through learned weighted factors. This unique capability of CSAM, combined with LFF's method of filtering the most influential features, enriches the feature extraction process. Additionally, the use of skip connections in our architecture ensures the retention of essential original features, culminating in a robust and efficient super-resolution framework.

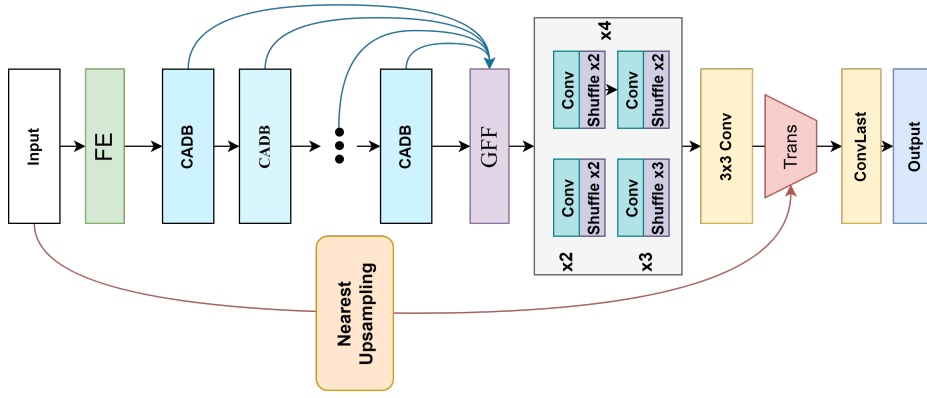
Let F_{d-1} and F_d be the input and output of the d^{th} CADB 3×3 convolution function and both of them have G_0 feature maps. The output of F_d can be formulated as

$$F_d = \sigma(W_d[F_{d-1}, \ldots, F_1]), \tag{11}$$

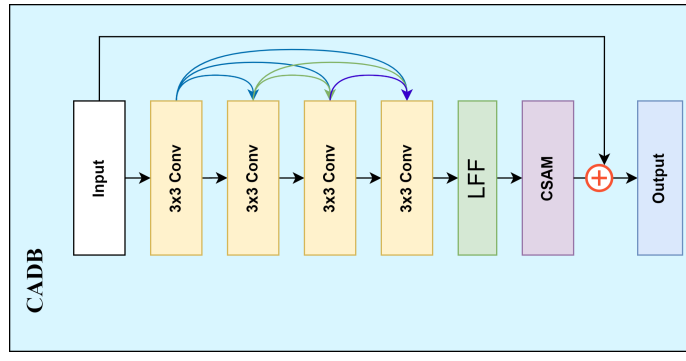
where σ denotes the activation function. W_d is the weight of the d^{th} Conv layer. $[F_{d-1}, \ldots, F_1]$ refer to the concatenation of feature maps produced by the previous Conv layers. Each Conv layer accepts the previous concatenation feature maps and outputs the same G_0 feature maps. The dense connections help every Conv layer connects to any previous Conv layer which results in a very deep feature extraction.

Local feature fusion is used to keep the output feature maps consistent with outside CADBs. Concatenate the previous d Conv layers feature maps and the last direct output of d^{th} Conv layer, which results in $(d+1)G_0$ feature maps. Apply the function of LFF, the output feature F_{LFF} can be expressed as

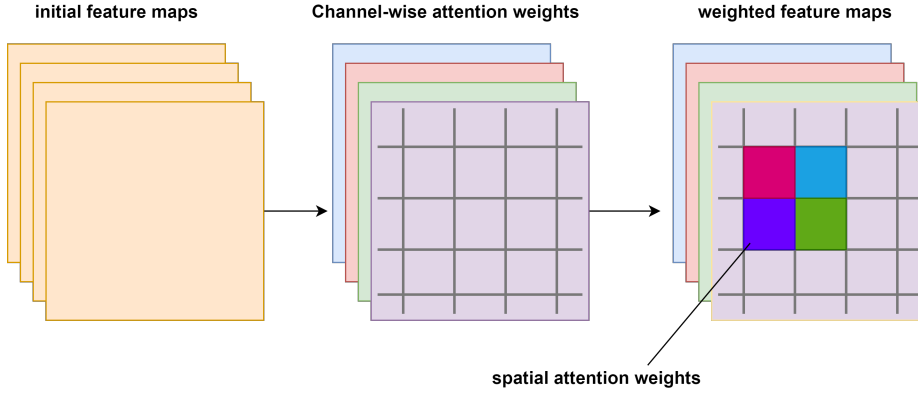
$$F_{LFF} = H_{LFF}([F_d, F_{d-1}, \ldots, F_1, F_D], \tag{12}$$



a) BATN



b) CADB



c) CSAM

Figure 2. Overview of the network architecture

where the H_{LFF} denotes the feature fusion convolution function, which is a 1×1 pointwise convolution. It mainly shrinks the concatenated feature map channels into a smaller number of channels. Using this feature fusion can prevent the network grow very huge and also can help filter out the redundancy feature maps.

The Channel-Spatial Attention Mechanism (CSAM) is shown in Figure 2 c). We use this CSAM to help us capture channel-wise and spatial-wise information. This mechanism is realized by two weight initialize functions Φ_c and Φ_s, using it we can get the channel descriptor α and spatial descriptor β. And then applying these weights to the input feature V, we get the weighted feature V_w as

$$\begin{aligned} \alpha &= \Phi_c(h_{t-1}, V), \\ \beta &= \Phi_s(h_{t-1}, f_c(V, \alpha)), \\ V_w &= f(V, \alpha, \beta). \end{aligned} \tag{13}$$

Specifically, to compute the spatial attention, we firstly apply average-pooling and max-pooling operations across the channels axis and then the pooled features will be concatenated and convolved by a standard convolution layer, finally by using an activate function to form the 2D spatial attention map. The spatial descriptor β can be formed as

$$\begin{aligned} \beta &= \Phi_s(V), \\ \beta &= \sigma(f^{7\times7}([AvgPool(V); MaxPool(V)])), \end{aligned} \tag{14}$$

where σ denotes the sigmoid function, $f^{7\times7}$ represents a convolution operation with the filter size of 7×7 and V denotes the input feature.

The channel attention map is a one dimension channel-wise descriptor $M_c \in \mathbb{R}^{C\times1\times1}$. We first apply average-pooling and max-pooling to aggregate the spatial information of a feature map, generating two one dimension spatial context descriptors: F^c_{avg} and F^c_{max}. Here c denotes then number of channels. And then both spatial context descriptors will be forwarded to a shared multi-layer perceptron (MLP) network, this shared network only contain one hidden layer with a activation size of $\mathbb{R}^{C/r\times1\times1}$, where r is the reduction ratio. After the forwarding of shared network, we merge the output feature vectors using element-wise summation and finally by using an activate function to form the 1D channel attention map. The channel descriptor α can be formed as

$$\begin{aligned} \alpha &= \Phi_c(V), \\ \alpha &= \sigma(MLP(AvgPool(V)) + MLP(MaxPool(V))), \end{aligned} \tag{15}$$

where σ denotes the sigmoid function, MLP denotes the shared multi-layer perceptron network and V denotes the input feature.

Finally, we add a skip connection from the input to the CSAM output, restoring some information lost and helping the gradient flow. The output of CADB F_{CADB}

expresses as

$$F_{CADB} = F_I + F_{CSAM}, \tag{16}$$

where F_I denotes the input of the CADB, F_{CSAM} denotes the output of CSAM.

3.3 Backbone Output Transform

As mentioned in Section 1, we try to perform a novel transform on the backbone output, which can directly improve the model performance without adding more expenditure. The most common transforms are

1. direct mapping the output to the target,
2. adding the original LR to form a residual connection,
3. multiplying the output with the original LR.

However, none of the studies seriously discussed the effects of these transforms and had a real exploration of these transforms. Here, we first use the nearest upsampling to upscale the LR input to the size of the backbone output, and then we let the backbone output and the upscaled LR input have some arithmetic operations between. The transformed output T_{trans} expresses as

$$T_{trans} = \mathbf{U}(I_{LR}) * T_{backbone}, \tag{17}$$

where $\mathbf{U}(I_{LR})$ denotes the upsampled LR input, $T_{backbone}$ denotes the backbone output and $*$ represents the arithmetic operations we use. And this is what we will explore in our paper, to see the details of final chosen $*$ operation, please refer to Table 2.

4 EXPERIMENTS

In this section, we present a comprehensive quantitative and qualitative evaluation of our approach. We first discuss the setup and datasets employed in our experiments. And then the detail of implementation and comparison with other methods are covered. Finally, we will discuss the effectiveness of our method in the ablation section.

4.1 Experimental Setup

Datasets and metrics. For training the proposed backbone target transform attention dense network, we employed the widely used DIV2K [34] image dataset. There are 1 000 high-quality images in the DIV2K dataset, where 800 images are for training, 100 images for validation and the other 100 images for testing. We select 785 images from DIV2K which the height and width is over 1024 pixels, this is because we use a large patch size in our experiment training. In the

testing, four standard datasets, i.e., the Set5 [35], Set14 [36], BSD100 [37] and Urban100 [38] were used as suggested by the EDSR paper [15]. Following the setting in [33], we evaluated the peak noise-signal ratio (PSNR) and SSIM [39] on the Y channel of images represented in the YCbCr (Y, Cb, Cr) colour space.

4.2 Implementation Details

Regarding the implementation of the BTAN network, we first increase the channels from 3 to 16 using a 3×3 convolution operation, which will keep our network small enough initially. And then again we use a 3×3 convolution layer to extract the shallow features, maintaining the 16 channels. We set the 8 CADBs in CADB group, and each CADB consists of 4 3×3 Conv layers. The channels of each input and output of the CADB are always 16, cause we use feature fusion to reduce the increased channels. Except the feature fusion layers use the 1×1 convolution operation, all the other convolution kernel is set to 3×3. The Trans layer is an arithmetic operation between input LR and the backbone output, which is $I_{LR} - (X/2 * I_{LR})$. We also have tested some other Trans operations which will be compared in Section 4.4.

Data augmentation is performed on the selected 785 training images, which are randomly rotated by 90°, 180°, 270° and flipped horizontally. We use a batch size of 16 and we use different patch sizes when we are training different scale models, which is 838×838, 558×558, 128×128 respectively to $2\times$, $3\times$, $4\times$ upscale models. And we will explain this in our Section 4.4. The entire framework was trained by ADAM optimizor with $\beta 1 = 0.9$, $\beta 2 = 0.999$ and $\epsilon = 10^{-8}$. We use L1 loss function to converge our model. The initial learning rate is set to 10^{-4} and then decreases to half every 2×10^5 iterations of back-propagation. We use BasicSR [40] to implement our models with two Nvidia 3090 GPUs. The source code of the proposed method can be downloaded at `https://github.com/philopatrick/BTAN`.

4.3 Comparison with State-of-the-Art Methods

In our evaluation of the proposed BTAN super-resolution models, we compared their performance to state-of-the-art SR methods using commonly-used image quality metrics, PSNR and SSIM. To represent the computational efficiency of the models, we used the MultAdds metric, which measures the number of composite multiply-accumulate operations for a single image, assuming the HR image size to be 720 p ($1\,280 \times 720$). Figure 3 presents a comparison between our BTAN and various benchmark algorithms on the Set14 $\times$ 4 dataset, based on the MultAdds and the number of parameters. At the PSNR span of 37.6 to 37.8, our BTAN has the best balance between model size and operations. Despite CARN, MemNet, DRRN all having better PSNR performance than us, we have a more compact model size and fewer operations yet achieve similar PSNR results. The CARN has slightly better performance than us, but it has more than 7 times the model size than us. The

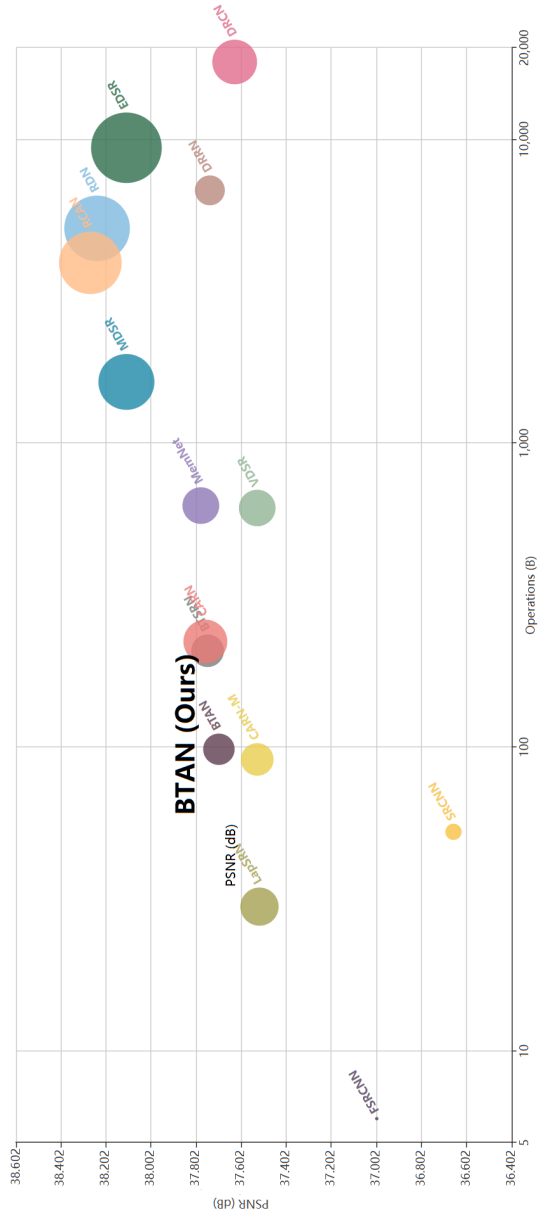


Figure 3. Trade-off between performance (measured in PSNR) and the number of operations and parameters used. The x-axis represents the number of operations (MultAdds), and the y-axis represents the PSNR. The size of each circle on the graph corresponds to the number of parameters used. The MultAdds value is calculated based on the assumption that the resolution of the high-resolution (HR) image is 720 p.

DRRN also has similar results to ours, but it has nearly 9 times of operations than ours.

The performance comparisons over the benchmark datasets are presented in Table 1. It firstly shows a difference of Params and MultAdds, typically the parameter size is correlated with the computation expense, the larger the parameter size, the larger the computation expense will be. However, some of the models which use recursive learning will have a small models size with a reversely large computation cost. The computation operation (MultAdds) value is calculated based on the assumption that the resolution of the high-resolution (HR) image is 720 p. Assumed the upsampled target image size is same, then those post-upsample models will have different input size with different upsample scales, and those pre-upsample models will always have the same input size because they will preprocess the input size to be the same size with the output size. With the different input size, there will be different computation operations to these post-upsample models, and because usually some models will change the upsample part when resolve different scales, it will also have some minor differences in model parameters among different scales.

Our comparative analysis in Table 1 focuses on recent lightweight SR models from the past two years, such as MOREMNAS-A, FALSR-C, AWSRN-S, and SplitSR. MemNet is excluded due to its significantly higher parameter count. Our BATN model, with only 213 k parameters for 2× and 223 k for 4× scaling, not only demonstrates superior performance over models like CARN-M but also shows remarkable efficiency. For instance, it outperforms FALSR-C in 2× scaling with fewer MultAdds, and in 4× scaling, it surpasses AWSRN-S and SplitSR with lower computational complexity and better PSNR and SSIM scores. Comparing with all the recently light-weight SR models, our model has the best or second-best on all the benchmark datasets in terms of PSNR and SSIM.

Figure 4 provides a visual representation of the qualitative comparisons for two datasets (Set14, Urban100) at a ×4 scale. The figure clearly demonstrates that our model outperforms the other models, as it is able to accurately reconstruct not only stripes and line patterns but also complex objects like hands.

4.4 Ablation Study

Here we evaluate some effective settings of our proposed method, which is using a bigger patch size, and adding the CSAM module and Trans module. A baseline model is designed to test these settings, we build it with our BTAN which removed the CSAM and Trans modules. It should be noted that in our CADB, the dense connection and the local feature fusion (LFF) are not evaluated as for their effectiveness here, because their effectiveness in super resolution are already proved by RDB [20].

Effects of different patch size. We use different patch sizes to train our baseline model, although the model and the training dataset remain the same. The

Scale	Model	Params	MultAdds	Set5 PSNR/SSIM	Set14 PSNR/SSIM	B100 PSNR/SSIM	Urban100 PSNR/SSIM
2	SRCNN [13]	57 k	52.7 G	36.66/0.9542	32.42/0.9063	31.36/0.8879	29.50/0.8946
	FSRCNN [16]	12 k	6.0 G	37.00/0.9558	32.63/0.9088	31.53/0.8920	29.88/0.9020
	VDSR [14]	665 k	612.6 G	37.53/0.9587	33.03/0.9124	31.90/0.8960	30.76/0.9140
	LapSRN [41]	813 k	29.9 G	37.52/0.9590	33.08/0.9130	31.80/0.8950	30.41/0.9100
	DRCN [17]	1 774 k	9 788.7 G	37.63/0.9588	33.04/0.9118	31.85/0.8942	30.75/0.9133
	CNF [42]	337 k	311.0 G	37.66/0.9590	33.38/0.9136	31.91/0.8962	–
	DRRN [18]	297 k	6 796.9 G	37.74/0.9591	33.23/0.9136	32.05/0.8973	31.23/0.9188
	CARN-M [31]	412 k	91.2 G	37.53/0.9583	33.26/0.9141	31.92/0.8960	30.83/0.9233
	MOREMNAS-A [43]	1 039 k	238.6 G	37.63/0.9584	33.23/0.9138	31.95/0.8961	31.24/0.9187
	FALSR-C [44]	408 k	93.7 G	37.66/0.9586	33.26/0.9140	31.96/0.8965	31.24/0.9187
	BATN (ours)	213 k	98.5 G	37.72/0.9596	33.25/0.9148	31.94/0.8967	31.24/0.9195
3	SRCNN [13]	57 k	52.7 G	32.75/0.9090	29.28/0.8209	28.41/0.7863	26.24/0.7989
	FSRCNN [16]	12 k	5.0 G	33.16/0.9140	29.43/0.8242	28.53/0.7910	26.43/0.8080
	VDSR [14]	665 k	612.6 G	33.66/0.9213	29.77/0.8314	28.82/0.7976	27.14/0.8279
	DRCN [17]	1 774 k	9 788.7 G	33.82/0.9226	29.76/0.8311	28.80/0.7963	27.15/0.8276
	CNF [42]	337 k	311.0 G	33.74/0.9226	29.90/0.8322	28.82/0.7980	–
	DRRN [18]	297 k	6 796.9 G	34.03/0.9244	29.96/0.8349	28.95/0.8004	27.53/0.8378
	CARN-M [31]	412 k	46.1 G	33.99/0.9236	30.08/0.8367	28.91/0.8000	26.86/0.8263
	BATN (ours)	225 k	46.7 G	34.04/0.9242	30.08/0.8374	28.90/0.8004	27.49/0.8384
4	SRCNN [13]	57 k	52.7 G	30.48/0.8628	27.49/0.7503	26.90/0.7101	24.52/0.7221
	FSRCNN [16]	12 k	4.6 G	30.71/0.8657	27.59/0.7535	26.98/0.7150	24.62/0.7280
	VDSR [14]	665 k	612.6 G	31.35/0.8838	28.01/0.7674	27.29/0.7251	25.18/0.7524
	DRCN [17]	1 774 k	9 788.7 G	31.53/0.8854	28.02/0.7670	27.23/0.7233	25.14/0.7510
	CNF [42]	337 k	311.0 G	31.55/0.8856	28.15/0.7680	27.32/0.7253	–
	DRRN [18]	297 k	6 796.9 G	31.68/0.8888	28.21/0.7720	27.38/0.7284	25.44/0.7638
	CARN-M [31]	412 k	32.5 G	31.92/0.8903	28.42/0.7762	27.44/0.7304	25.63/0.7688
	AWSRN-S [45]	588 k	33.7 G	31.77/0.8893	28.35/0.7761	27.41/0.7304	25.56/0.7678
	SplitSR [46]	174 k	–	31.76/0.8882	28.29/0.7716	27.39/0.7291	25.46/0.769
	BATN (ours)	223 k	26.6 G	31.92/0.8909	28.38/0.7761	27.41/0.7314	25.56/0.7708

Table 1. Quantitative results of deep learning-based SR algorithms. Red/Blue text: best/second-best.

comparison of training with different patch sizes is shown in Figure 5. We found that using a bigger patch size will increase the model performance. Although using a bigger patch size also requires larger device memory to run the training, there will be a sweet point to training our model. By observing the Set14 line in the graph, we found using a patch size of 128 is economic in terms of performance and memory cost. And also conducted like this experiment, we tested that the economic points of training 2 times and 3 times are 838 and 558.

The reason why enlarging the patch size will improve the performance is that a bigger input patch will increase the possibility of capturing the context information. However, a model's capacity to process the information is limited, if we increase too much the patch size, there will be a drop in performance. Because the context information and detail information serve as opposites sometimes, if you take care of too much context, you will lose the detail and vice versa.

Effects of CSAM and Trans. Table 2 shows the ablation study of CSAM and different Trans. From the result, we can tell that using some Trans is better than not using any operation, and Trans operation used by BTAN-NCA is the outstanding one which has a margin of 0.22 PSNR improvement. And also the CSAM module is effective to have a margin of 0.04 improvement over the BTAN-NCA.

Using the equations like Equations (3) and (5), we can easily reversely deduct the corresponding backbone output of different Trans operations. Suppose the model output ideally converges with the ground truth, and then we can easily visualize these backbone outputs like Figure 6 shows. We can see that by using some Trans operations, the visualization of backbone output becomes much more

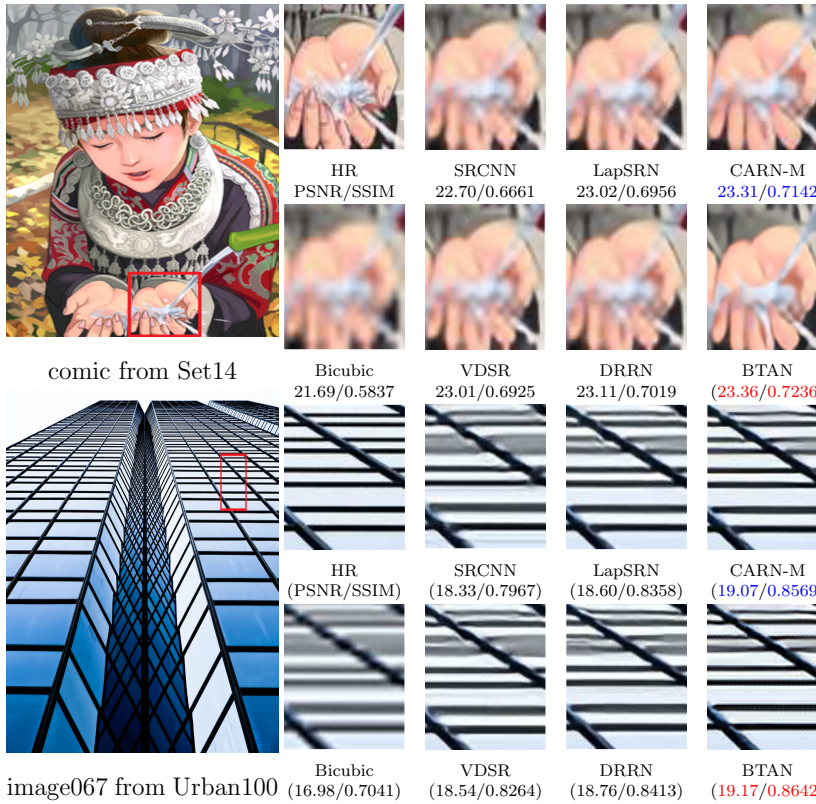
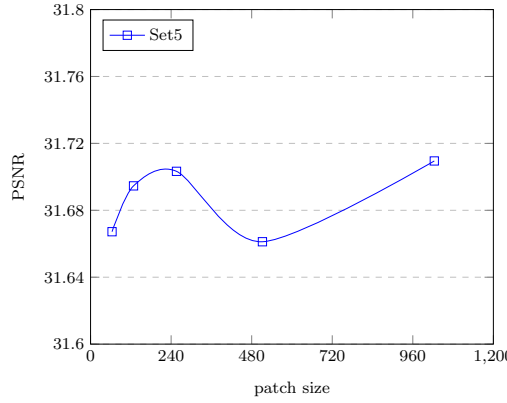


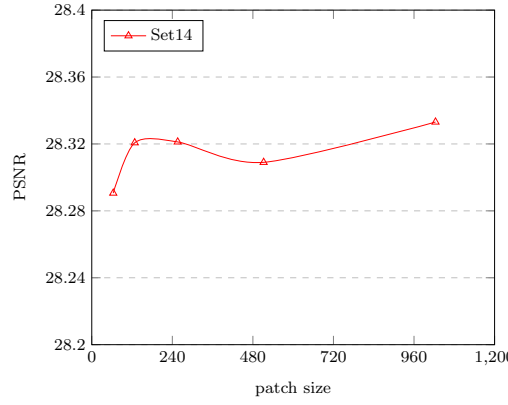
Figure 4. Visual qualitative comparison on ×4 scale datasets

	Baseline	Base-multi	Base-multiadd	BTAN-NCA	BTAN
pure	✓				
$I_{LR} * T_{backbone}$		✓			
$(I_{LR} * T_{backbone}) + I_{LR}$			✓		
$I_{LR} - (T_{backbone}/2 * I_{LR})$				✓	✓
CSAM					✓
PSNR	31.66	31.76	31.80	31.88	31.92

Table 2. Effects of the CSAM and Trans modules measured on the Set5 ×4 dataset. BTAN-NCA represents BTAN without CSAM. Base-multi represents Baseline model using operation of $I_{LR} * X$ and Base-multiadd represents using operation of $(I_{LR} * X) + I_{LR}$, where I_{LR} denotes LR input and X denotes the backbone ouput. Pure means no Trans module, the backbone output directly outputs to the result.



a) Tested on Set5 with 4× scale



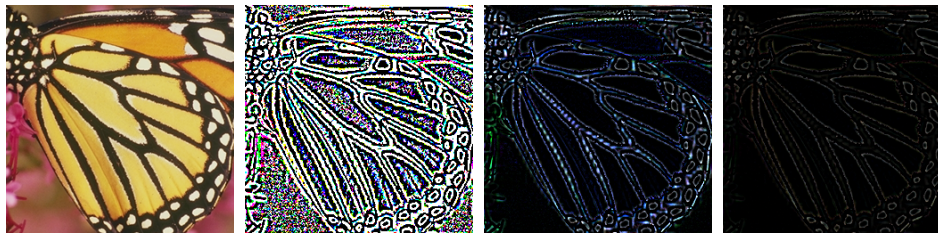
b) Tested on Set14 with 4× scale

Figure 5. Comparision of using different patch size when training model. Tested on Set5 and Set14 with 4× scale.

noise free. That means we lessen the burden of the backbone network and help that it can better finish the "task" assigned to it.

5 CONCLUSION

In this paper, we proposed a new approach to improve the performance of lightweight image super-resolution networks without increasing the number of parameters. Our approach focuses on exploring the relationship between low-resolution images and high-resolution images, rather than simply optimizing the network structure itself. We found that this approach can significantly enhance the performance of lightweight networks, achieving state-of-the-art results.



a) Trans using none b) Trans using add c) Trans using multiadd d) BTAN Trans

Figure 6. Visualization of the backbone output under different Trans operation

Moreover, we examined the effect of using different patch sizes in training a super-resolution network. Our experiments showed that using larger patch sizes can improve the accuracy of the super-resolution model, but this comes at the cost of increased computational complexity.

Based on our findings, we developed a novel network architecture, BTAN, that incorporates our proposed approach and achieves state-of-the-art performance on a variety of benchmark datasets. BTAN is not only lightweight but also highly efficient, making it a promising solution for real-world applications that require fast and accurate image super-resolution.

Although our approach has shown promising results in improving the performance of lightweight image super-resolution networks, it is important to consider its limitations. Firstly, our method focuses on exploring the relationship between low-resolution images and high-resolution images rather than optimizing the network structure itself. While this approach has proven effective, there may be alternative network architectures or additional information sources that could further enhance performance. Future research should explore these possibilities to push the boundaries of image super-resolution.

Secondly, it is important to note that our method is primarily designed for resource-constrained devices such as mobile phones or embedded systems. The applicability of our approach to high-performance computing environments or large-scale image processing tasks may require further investigation. It would be valuable to assess how well our method performs in these scenarios and whether any modifications are necessary.

Additionally, we discussed the trade-off between accuracy and computational efficiency when using larger patch sizes during training. However, other factors such as the choice of loss function or data augmentation techniques could also impact the model's performance. These aspects should be carefully considered when implementing our method in practical applications.

In conclusion, our research contributes to the field of image super-resolution by providing a new perspective on this topic and offering a practical solution that can address the challenges of resource-intensive deep neural networks. Our findings have important implications for a wide range of applications, including medical

imaging, remote sensing, and surveillance systems. Future work could explore the generalizability of our approach to other image-processing tasks and investigate the potential of combining it with other techniques, such as attention mechanisms and adversarial training.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No. 62176217) and the Innovation Team Funds of China West Normal University (Grant No. KCXTD2022-3).

REFERENCES

[1] Yang, W.—Zhang, X.—Tian, Y.—Wang, W.—Xue, J. H.—Liao, Q.: Deep Learning for Single Image Super-Resolution: A Brief Review. IEEE Transactions on Multimedia, Vol. 21, 2019, No. 12, pp. 3106–3121, doi: 10.1109/TMM.2019.2919431.

[2] Aakerberg, A.—Johansen, A. S.—Nasrollahi, K.—Moeslund, T. B.: Semantic Segmentation Guided Real-World Super-Resolution. 2022 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW), 2022, pp. 449–458, doi: 10.1109/WACVW54805.2022.00051.

[3] Ayazoglu, M.: Extremely Lightweight Quantization Robust Real-Time Single-Image Super Resolution for Mobile Devices. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 2472–2479, doi: 10.1109/CVPRW53098.2021.00280.

[4] Chen, X.—Wang, X.—Zhou, J.—Qiao, Y.—Dong, C.: Activating More Pixels in Image Super-Resolution Transformer. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 22367–22377, doi: 10.1109/CVPR52729.2023.02142.

[5] Hao, F.—Ma, X.—Zhang, T.—Tang, Y.: Channel Hourglass Residual Network for Single Image Super-Resolution. 2021 International Joint Conference on Neural Networks (IJCNN), 2021, pp. 1–8, doi: 10.1109/IJCNN52387.2021.9533568.

[6] Zhang, J.—Qu, Y.—Chen, L.: Using Conv-LSTM to Refine Features for Lightweight Image Super-Resolution Network. Image and Graphics (ICIG 2021), Springer, Cham, Lecture Notes in Computer Science, Vol. 12890, 2021, pp. 230–240, doi: 10.1007/978-3-030-87361-5_19.

[7] Zhao, H.—Kong, X.—He, J.—Qiao, Y.—Dong, C.: Efficient Image Super-Resolution Using Pixel Attention. Computer Vision – ECCV 2020 Workshops, Springer, Cham, Lecture Notes in Computer Science, Vol. 12537, 2020, pp. 56–72, doi: 10.1007/978-3-030-67070-2_3.

[8] Kong, F.—Li, M.—Liu, S.—Liu, D.—He, J.—Bai, Y.—Chen, F.—Fu, L.: Residual Local Feature Network for Efficient Super-Resolution. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2022, pp. 765–775, doi: 10.1109/CVPRW56347.2022.00092.

[9] Shang, T.—Dai, Q.—Zhu, S.—Yang, T.—Guo, Y.: Perceptual Extreme Super Resolution Network with Receptive Field Block. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 1778–1787, doi: 10.1109/CVPRW50498.2020.00228.

[10] Yang, F.—Yang, H.—Fu, J.—Lu, H.—Guo, B.: Learning Texture Transformer Network for Image Super-Resolution. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 5790–5799, doi: 10.1109/CVPR42600.2020.00583.

[11] Lu, Z.—Li, J.—Liu, H.—Huang, C.—Zhang, L.—Zeng, T.: Transformer for Single Image Super-Resolution. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2022, pp. 456–465, doi: 10.1109/CVPRW56347.2022.00061.

[12] Liang, J.—Cao, J.—Sun, G.—Zhang, K.—Van Gool, L.—Timofte, R.: SwinIR: Image Restoration Using Swin Transformer. 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), 2021, pp. 1833–1844, doi: 10.1109/ICCVW54120.2021.00210.

[13] Dong, C.—Loy, C. C.—He, K.—Tang, X.: Image Super-Resolution Using Deep Convolutional Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 38, 2016, No. 2, pp. 295–307, doi: 10.1109/TPAMI.2015.2439281.

[14] Kim, J.—Lee, J. K.—Lee, K. M.: Accurate Image Super-Resolution Using Very Deep Convolutional Networks. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1646–1654, doi: 10.1109/CVPR.2016.182.

[15] Lim, B.—Son, S.—Kim, H.—Nah, S.—Lee, K. M.: Enhanced Deep Residual Networks for Single Image Super-Resolution. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 136–144, doi: 10.1109/CVPRW.2017.151.

[16] Dong, C.—Loy, C. C.—Tang, X.: Accelerating the Super-Resolution Convolutional Neural Network. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (Eds.): Computer Vision – ECCV 2016. Springer, Cham, Lecture Notes in Computer Science, Vol. 9906, 2016, pp. 391–407, doi: 10.1007/978-3-319-46475-6_25.

[17] Kim, J.—Lee, J. K.—Lee, K. M.: Deeply-Recursive Convolutional Network for Image Super-Resolution. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1637–1645, doi: 10.1109/CVPR.2016.181.

[18] Tai, Y.—Yang, J.—Liu, X.: Image Super-Resolution via Deep Recursive Residual Network. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2790–2798, doi: 10.1109/CVPR.2017.298.

[19] Tong, T.—Li, G.—Liu, X.—Gao, Q.: Image Super-Resolution Using Dense Skip Connections. 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 4809–4817, doi: 10.1109/ICCV.2017.514.

[20] Zhang, Y.—Li, K.—Li, K.—Wang, L.—Zhong, B.—Fu, Y.: Image Super-Resolution Using Very Deep Residual Channel Attention Networks. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (Eds.): Computer Vision – ECCV 2018. Springer, Cham, Lecture Notes in Computer Science, Vol. 11211, 2018, pp. 294–310, doi: 10.1007/978-3-030-01234-2_18.

[21] Zhang, Y.—Tian, Y.—Kong, Y.—Zhong, B.—Fu, Y.: Residual Dense Network for Image Super-Resolution. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 2472–2481, doi: 10.1109/CVPR.2018.00262.

[22] Kong, S.—Fowlkes, C.: Image Reconstruction with Predictive Filter Flow. CoRR, 2018, doi: 10.48550/arXiv.1811.11482.

[23] Iandola, F.—Moskewicz, M.—Karayev, S.—Girshick, R.—Darrell, T.—Keutzer, K.: DenseNet: Implementing Efficient ConvNet Descriptor Pyramids. CoRR, 2014, doi: 10.48550/arXiv.1404.1869.

[24] Chen, L.—Zhang, H.—Xiao, J.—Nie, L.—Shao, J.—Liu, W.—Chua, T. S.: SCA-CNN: Spatial and Channel-Wise Attention in Convolutional Networks for Image Captioning. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6298–6306, doi: 10.1109/CVPR.2017.667.

[25] Goodfellow, I.—Pouget-Abadie, J.—Mirza, M.—Xu, B.—Warde-Farley, D.—Ozair, S.—Courville, A.—Bengio, Y.: Generative Adversarial Networks. Communications of the ACM, Vol. 63, 2020, No. 11, pp. 139–144, doi: 10.1145/3422622.

[26] Ledig, C.—Theis, L.—Huszár, F.—Caballero, J.—Cunningham, A.—Acosta, A.—Aitken, A.—Tejani, A.—Totz, J.—Wang, Z.—Shi, W.: Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 105–114, doi: 10.1109/CVPR.2017.19.

[27] Wang, X.—Yu, K.—Wu, S.—Gu, J.—Liu, Y.—Dong, C.—Qiao, Y.—Change Loy, C.: ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. In: Leal-Taixé, L., Roth, S. (Eds.): Computer Vision – ECCV 2018 Workshops. Springer, Cham, Lecture Notes in Computer Science, Vol. 11133, 2018, pp. 63–79, doi: 10.1007/978-3-030-11021-5_5.

[28] Shi, W.—Caballero, J.—Huszár, F.—Totz, J.—Aitken, A. P.—Bishop, R.—Rueckert, D.—Wang, Z.: Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1874–1883, doi: 10.1109/CVPR.2016.207.

[29] Tai, Y.—Yang, J.—Liu, X.—Xu, C.: MemNet: A Persistent Memory Network for Image Restoration. 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 4549–4557, doi: 10.1109/ICCV.2017.486.

[30] Ahn, N.—Kang, B.—Sohn, K. A.: Fast, Accurate, and Lightweight Super-Resolution with Cascading Residual Network. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (Eds.): Computer Vision – ECCV 2018. Springer, Cham, Lecture Notes in Computer Science, Vol. 11214, 2018, pp. 256–272, doi: 10.1007/978-3-030-01249-6_16.

[31] Hui, Z.—Wang, X.—Gao, X.: Fast and Accurate Single Image Super-Resolution via Information Distillation Network. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 723–731, doi: 10.1109/CVPR.2018.00082.

[32] Lu, Z.—Li, J.—Liu, H.—Huang, C.—Zhang, L.—Zeng, T.: Transformer for Single Image Super-Resolution. 2022 IEEE/CVF Conference on Computer

Vision and Pattern Recognition Workshops (CVPRW), 2022, pp. 456–465, doi: 10.1109/CVPRW56347.2022.00061.

[33] Zhou, L.—Cai, H.—Gu, J.—Li, Z.—Liu, Y.—Chen, X.—Qiao, Y.—Dong, C.: Efficient Image Super-Resolution Using Vast-Receptive-Field Attention. In: Karlinsky, L., Michaeli, T., Nishino, K. (Eds.): Computer Vision – ECCV 2022 Workshops. Springer, Cham, Lecture Notes in Computer Science, Vol. 13802, 2022, pp. 256–272, doi: 10.1007/978-3-031-25063-7_16.

[34] Agustsson, E.—Timofte, R.: NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017, pp. 1122–1131, doi: 10.1109/CVPRW.2017.150.

[35] Bevilacqua, M.—Roumy, A.—Guillemot, C.—Alberi-Morel, M. L.: Low-Complexity Single-Image Super-Resolution Based on Nonnegative Neighbor Embedding. In: Bowden, R., Collomosse, J., Mikolajczyk, K. (Eds.): Proceedings of the 23rd British Machine Vision Conference (BMVC). BMVA Press, 2012, https://bmva-archive.org.uk/bmvc/2012/BMVC/paper135/paper135.pdf.

[36] Zeyde, R.—Elad, M.—Protter, M.: On Single Image Scale-Up Using Sparse-Representations. In: Boissonnat, J. D., Chenin, P., Cohen, A., Gout, C., Lyche, T., Mazure, M. L., Schumaker, L. (Eds.): 7th International Conference on Curves and Surfaces (Curves and Surfaces 2010). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 6920, 2010, pp. 711–730, doi: 10.1007/978-3-642-27413-8_47.

[37] Martin, D.—Fowlkes, C.—Tal, D.—Malik, J.: A Database of Human Segmented Natural Images and Its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. Proceedings Eighth IEEE International Conference on Computer Vision (ICCV 2001), IEEE, Vol. 2, 2001, pp. 416–423, doi: 10.1109/ICCV.2001.937655.

[38] Huang, J. B.—Singh, A.—Ahuja, N.: Single Image Super-Resolution from Transformed Self-Exemplars. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 5197–5206, doi: 10.1109/CVPR.2015.7299156.

[39] Wang, Z.—Bovik, A. C.—Sheikh, H. R.—Simoncelli, E. P.: Image Quality Assessment: From Error Visibility to Structural Similarity. IEEE Transactions on Image Processing, Vol. 13, 2004, No. 4, pp. 600–612, doi: 10.1109/TIP.2003.819861.

[40] Wang, X.—Xie, L.—Yu, K.—Chan, K. C.—Loy, C. C.—Dong, C.: BasicSR: Open Source Image and Video Restoration Toolbox. 2022, https://github.com/XPixelGroup/BasicSR.

[41] Lai, W. S.—Huang, J. B.—Ahuja, N.—Yang, M. H.: Deep Laplacian Pyramid Networks for Fast and Accurate Super-Resolution. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 5835–5843, doi: 10.1109/CVPR.2017.618.

[42] Liu, D.—Wen, B.—Fan, Y.—Loy, C. C.—Huang, T. S.: Non-Local Recurrent Network for Image Restoration. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (Eds.): Advances in Neural Information Processing Systems 31 (NeurIPS 2018). Curran Associates, Inc., 2018, pp. 1673–1682, doi: 10.48550/arXiv.1806.02919.

[43] Chu, X.—Zhang, B.—Xu, R.: Multi-Objective Reinforced Evolution in Mobile Neural Architecture Search. Computer Vision – ECCV 2020 Workshops, Springer, Cham, Lecture Notes in Computer Science, Vol. 12538, 2020, pp. 99–113, doi: 10.1007/978-3-030-66823-5_6.

[44] Chu, X.—Zhang, B.—Ma, H.—Xu, R.—Li, Q.: Fast, Accurate and Lightweight Super-Resolution with Neural Architecture Search. 2020 25th Conference on Pattern Recognition (ICPR), IEEE, 2021, pp. 59–64, doi: 10.1109/ICPR48806.2021.9413080.

[45] Wang, C.—Li, Z.—Shi, J.: Lightweight Image Super-Resolution with Adaptive Weighted Learning Network. CoRR, 2019, doi: 10.48550/arXiv.1904.02358.

[46] Liu, X.—Li, Y.—Fromm, J.—Wang, Y.—Jiang, Z.—Mariakakis, A.—Patel, S.: SplitSR: An End-to-End Approach to Super-Resolution on Mobile Devices. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, Vol. 5, 2021, No. 1, Art. No. 25, doi: 10.1145/3448104.

Pan Wang is a postgraduate student in the School of Computer Science, at China West Normal University. His main interest is image super-resolution.

Zedong Wu is a postgraduate student in the School of Computer Science, at China West Normal University. His main interest is object detection.

Zicheng Ding is a postgraduate student in the School of Computer Science, at China West Normal University. His main interest is image restoring.

Bochuan Zheng received his Ph.D. in computer science from the University of Electronic Science and Technology of China, China, in 2012, and his M.Sc. in computer science from the Zhejiang University, China, in 2004. Currently, he has been working as Professor at the School of Computer Science, China West Normal University, China since 1998. His research interests include deep learning and computer vision.