

# Training Binary Neural Network without Batch Normalization for Image Super-Resolution

Xinrui Jiang<sup>1</sup>, Nannan Wang<sup>1\*</sup>, Jingwei Xin<sup>2</sup>, Keyu Li<sup>1</sup>, Xi Yang<sup>1</sup>, Xinbo Gao<sup>3</sup>

<sup>1</sup>State Key Laboratory of Integrated Services Networks,  
School of Telecommunications Engineering, Xidian University, Xi'an 710071, China

<sup>2</sup>State Key Laboratory of Integrated Services Networks,  
School of Electronic Engineering, Xidian University, Xi'an 710071, China

<sup>3</sup>Chongqing Key Laboratory of Image Cognition,  
Chongqing University of Posts and Telecommunications, Chongqing 400065, China

xrjiang@stu.xidian.edu.cn, {nnwang,yangx}@xidian.edu.cn, {jwxintt,likeyuchn}@gmail.com, gaobx@cqupt.edu.cn

## Abstract

Recently, binary neural network (BNN) based super-resolution (SR) methods have enjoyed initial success in the SR field. However, there is a noticeable performance gap between the binarized model and the full-precision one. Furthermore, the batch normalization (BN) in binary SR networks introduces floating-point calculations, which is unfriendly to low-precision hardware. Therefore, there is still room for improvement in terms of model performance and efficiency. Focusing on this issue, in this paper, we first explore a novel binary training mechanism based on the feature distribution, allowing us to replace all BN layers with a simple training method. Then, we construct a strong baseline by combining the highlights of recent binarization methods, which already surpasses the state-of-the-arts. Next, to train highly accurate binarized SR model, we also develop a lightweight network architecture and a multi-stage knowledge distillation strategy to enhance the model representation ability. Extensive experiments demonstrate that the proposed method not only presents advantages of lower computation as compared to conventional floating-point networks but outperforms the state-of-the-art binary methods on the standard SR networks.

## Introduction

Single image super-resolution (SR), which aims to reconstruct a high-resolution (HR) image from its degraded low-resolution (LR) version, has gained increasing research attention for decades. It enjoys various applications ranging from medical imaging (Greenspan 2008) to security and surveillance imaging (Rasti et al. 2016; Xin et al. 2020b). In general, image super-resolution is an ill-posed problem due to the one-to-many mapping. To address this inverse problem, various deep learning-based methods have been proposed and achieved significant success.

With the development of deep neural networks, "Bigger is better" becomes the ruling maxim in deep learning

land. Motivated by this, one major trend of SR research is increasing convolution layers to improve performance (Lim et al. 2017; Zhang et al. 2018b,a). Although these models receive excellent performance, they all require large amounts of computational sources, which severely restrict their applications in the real world. Therefore, there have been imperative demands for light-weight and efficient CNNs for real-world applications. To achieve this, many researchers begin to explore the fewer parameters and less computational complexity model architecture to improve the effectiveness of networks, such as recursive learning and parameters sharing (DRRN (Tai, Yang, and Liu 2017) and DRCN (Kim, Kwon Lee, and Mu Lee 2016b)), squeeze operation (e.g., IDMN (Hui et al. 2019) and ESRN (Song et al. 2020)), group convolution (e.g., CARN-M (Ahn, Kang, and Sohn 2018)) and wavelet domain (Xin et al. 2020a). Nevertheless, these approaches heavily rely on high-performance hardware such as GPU, which are still a huge burden for smaller devices like cell phones.

As a method to significantly reduce the model size and inferring time, model quantization recently has been introduced into SR field and has achieved initial success in BNN-based SR field. The binary neural network (BNN), where the full-precision floating-point weights and activations are replaced with 1-bit representations, enjoys  $32\times$  memory compression ratio and  $58\times$  speed up on CPUs (Rastegari et al. 2016). Benefiting from it, Xin et al. (Xin et al. 2020c) first explored a complete binary SR network, which utilizes a bit-accumulation mechanism (BAM) to approximate the full-precision convolution.

For a binary network, the input activations and weights can only choose two large-scale values  $\{-1, +1\}$ , exploding feature map values and making it hard to learn a good feature distribution. To alleviate this problem, following the BNN-based image classification methods, BAM also adopts batch normalization (BN) operations to improve the representation ability of feature maps. By re-centering the input distribution, the network could mitigate the effects of the quantization process and advance performance. However, BN introduces full-precision floating-point calculations and is un-

\*Corresponding author: Nannan Wang  
(nnwang@xidian.edu.cn)

Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

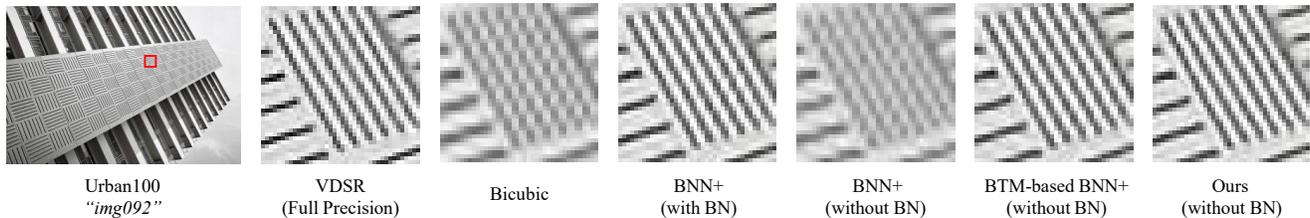


Figure 1: Visual qualitative comparison on binarized networks based on VDSR

friendly to low-precision hardware, resulting in the model less efficient. Besides, there is still a noticeable performance gap between the binarized model and the full-precision one.

Considering the above problems, in this paper, our goal is to train highly accurate binarized SR models without BN layers. The main contributions of our paper are:

- We explore an effective binary training mechanism (BTM) based on the feature distribution, which helps binary SR networks obtain excellent performance without BN layers (shown in Fig. 1). It means our method further saves model storage and computation, which is more energy-efficient for low-precision devices.
- We perform a thorough experiment to find the most well-suited optimization method and construct a strong baseline by combining the highlights of recent binarization methods. The experimental results show that our baseline already surpasses all previously binarization methods.
- A novel binary network architecture and a multi-stage knowledge distillation strategy are proposed to further improve the representation ability of binary SR networks. Equipped with these, compared to the state-of-the-arts, our method achieves a significant performance gain.

## Related Work

**Single Image Super-Resolution** Nowadays various SR methods have been proposed ranging from early classical methods (Zhang et al. 2014) to recent promising deep learning-based methods (Zhang and Tao 2020). Among them, SRCNN (Dong et al. 2015) contains only three convolutional layers, which is the first deep learning-based SR network. Later on, VDSR (Kim, Kwon Lee, and Mu Lee 2016a) increases the network depth to 20 layers and introduces residual learning to alleviate the vanishing-gradient problem. EDSR (Lim et al. 2017) adopts the residual block and extremely expands the network depth, which dramatically advances the SR performance. Recently, RCAN (Zhang et al. 2018a) utilizes the long and short residual skip connections and channel attention strategy to obtain a very deep SR network. CS-NL (Mei et al. 2020) develops a cross-scale non-local attention network to improve the network performance. Besides extensive efforts spent on improving SR performance by expanding network size and attention strategy, lightweight SR models also have been explored. CARN (Ahn, Kang, and Sohn 2018) designs an architecture that implemented a cascading mechanism on a residual network and the version CARN-M adopts group convolution to

reduce model parameters and computational complexity. In addition, IDMN (Hui et al. 2019) adopts a lightweight information multi-distillation network for image SR. OISR (He et al. 2019) utilizes ODE-inspired schemes (Chen et al. 2018) to alleviate the huge amount of calculations in SR processing. Recently, Xin et al. (Xin et al. 2020c) first explored to binarize the weights and activations of SR networks, which also utilized a bit-accumulation mechanism to approximate the full-precision convolution.

**Binary Neural Networks** Neural network binarization originates from BNN (Hubara et al. 2016), where weights and activations are restricted to +1 or -1. Later on, XNOR-Net (Rastegari et al. 2016) improves the performance of BNN by introducing a gain term to compensate for lost information. In recently, Bi-Real (Liu et al. 2018) modifies the ResNet architecture in the forward propagation and introduced a customized Approxsign function to compute the gradient in the backward propagation. Similarly, BNN+ (Darabi et al. 2018) adopts an improved approximation based on linear operations of the derivative of the sign activation function in the backward propagation. Besides, RTN (Li et al. 2020) reparameterizes the weights and activations with scale and offset parameters. IRNet (Qin et al. 2020) develops a novel information retention network to retain the information in the forward and backward propagation.

## Methodology

### Binary Training Mechanism

In general, most existing binarization strategies do not modify the backbone architecture, which usually binarize residual blocks to train models. Let  $I_n$  denote the input activations and  $I_{n+1}$  denote the output activations. For a binary convolution operation, there is a widely-used setting (shown in Fig 2a):

$$I_{n+1} = BN(F_A(C_B(A_B(I_n)))) \quad (1)$$

where  $A_B$  denotes the binarization process for input activations,  $C_B$  denotes the binary convolution,  $F_A$  and  $BN$  are the activation function and batch normalization.

As shown in Eq. (1), after the full-precision convolution is replaced with binary convolution, BN becomes the highest computational complexity operation. It introduces full-precision floating-point calculations and is unfriendly to low-precision hardware. As shown in Fig. 5, we could find that removing BN layers directly from the BSR network will seriously degrade network performance.

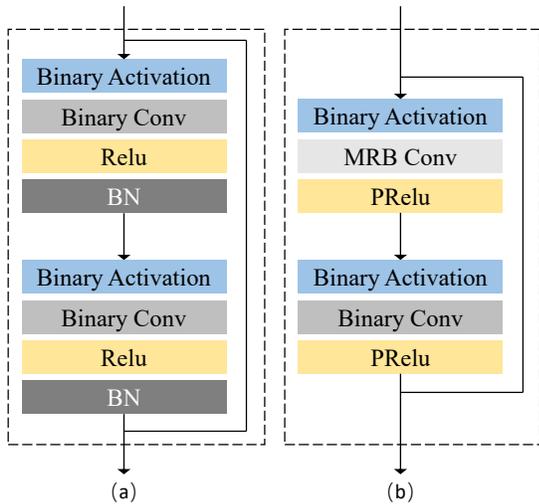


Figure 2: (a) original binary residual block (b) the proposed multiple receptive-field binary residual block (MRB)

To address this problem, we first consider the effect of BN on binary networks. As we know, BN layers are critical components for mitigating the 1-bit effects by re-centering the input distribution. For a binary network, the output activations of BN layers will be limited to  $\{-1, +1\}$  in the subsequent convolution layer, while are independent of its magnitude. Therefore, the binary output from a symmetric feature distribution may be similar to that with BN. Besides, the performance degradation may be largely due to the lacking of BN in the first and last full-precision convolution operations, especially for the first convolution since the global residual learning is necessary for image-to-image translation task. According to the discussion above, we explore a new binary training mechanism (BTM) for better training BSR networks without BN layers. Details are as follows:

- **Weight initialization** In the training processing, 1-bit CNNs are trained with gradient-based learnable quantization methods. When without BN layers, a nonuniformity initialization such as min-max initialization makes the activation distribution toward one direction (greater than 0 or less than 0), resulting in the inadequate network training. Therefore, for binary convolutions, we use a simple initialization scheme: Xavier uniform distribution (Glorot and Bengio 2010), which is more stable and could achieve excellent performance.
- **Data initialization** For binarized networks, in the backward propagation, the gradient becomes zeros when entering the saturating zone (shown in Fig. 3). Therefore, it cancels the most gradients when inputs get too large, severely affecting the training of shallow networks. To make the training more stable and mitigate the effect of input magnitude, we propose to normalize the input of SR networks:

$$x = \frac{x - \bar{x}}{\sigma} \quad (2)$$

where  $x$  denotes the input images,  $\bar{x}$  and  $\sigma$  denote the

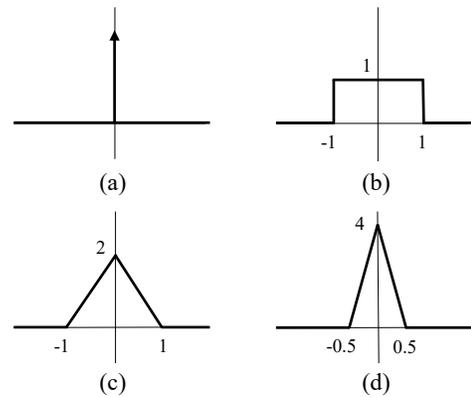


Figure 3: The derivative of sign function and different approximations for it

mean value and standard deviation from the dataset. Besides, we add the mean value from the output images in the last layer. The use of this makes the network obtain more stable training and reach a better optimization.

- **Activation** We introduce PReLU, which has a great ability of facilitating the binary network training.

### Baseline

Little binarized SR networks attempt to binarize both weights and activations in neural networks. Motivated by prior work, in this section, we explore recent advanced binary methods and construct a strong SR baseline by the following insights and settings:

**Network Settings:** Firstly, consistent with the existing work (Xin et al. 2020c), our network also adopts full-precision convolution in the first and last convolutional layers, and all other convolutional layers are binary convolution.

**Block Structure:** For a binary block, based on our proposed BTM, we adopt a new setting: Binary Activation  $\rightarrow$  Binary Convolution  $\rightarrow$  PReLU Function.

**Optimization:** In forward propagation, we minimize the quantization error by introducing the scaling factors proposed in (Rastegari et al. 2016). In backward propagation, due to the sign function is not differentiable (Fig. 3a), it cannot be applied for parameter updating directly. To handle this problem, BAM (Xin et al. 2020c) adopts STE (Bengio and Courville 2013) (Fig. 3b) for gradient computation in backward propagation, which causes a larger approximation error. In our baseline, to reduce the gradient mismatch between the gradient of Sign and STE, the higher-order estimators (Liu et al. 2018; Xu and Cheung 2019) are incorporated in our networks as the activation and weight approximation functions to guide gradient update (Fig. 3c and Fig. 3d).

Equipped with these, our baseline achieves excellent performance and already surpasses all previously binarization methods.

### Improved Binary Super-Resolution Network

**Network Structure** In this section, we further develop a novel binary architecture to improve the representation abil-

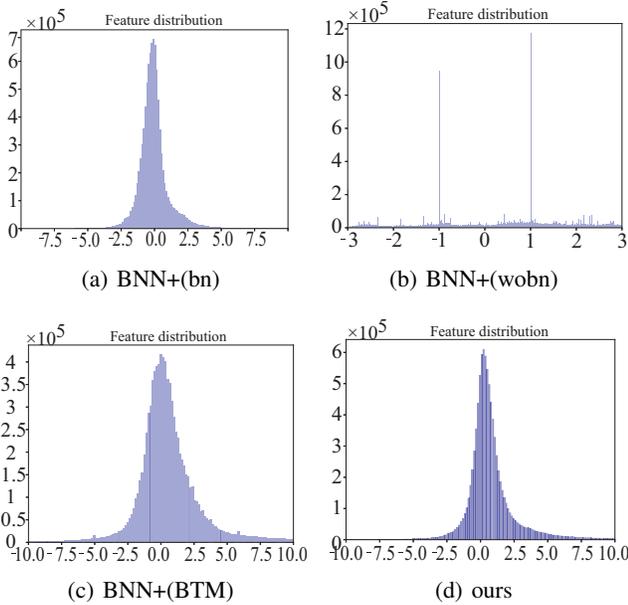


Figure 4: The output feature distribution of 4-th binary residual block in BNN+ based VDSR and ours. BTM denotes our proposed binary training mechanism

ity of binary feature maps. As multi-path learning is effectively in SR, we propose a multiple receptive-field block (MRB) to ease the information loss in the forward propagation. The basic building block is shown in Fig. 2b. The unit first learns the representations in parallel using dilated convolutions with different dilation rates. In order to learn a large receptive field without additional computational cost, we adopt the  $1 \times 5$  and  $5 \times 1$  convolution operation. Therefore, the first binary convolution operation in MRB can be expressed as:

$$I_{n+1} = F_A(\text{cat}(C_{B_{1 \times 5}}^1(A_B(I_n)), C_{B_{1 \times 5}}^2(A_B(I_n)), C_{B_{5 \times 1}}^1(A_B(I_n)), C_{B_{5 \times 1}}^2(A_B(I_n)))) \quad (3)$$

where  $C_{B_{m \times n}}^r$  denotes a  $m \times n$  binary dilation convolution with dilation rate  $r$  and  $\text{cat}(\cdot)$  is the cross-channel concatenated operation. To make it more computationally efficient, the output channels of  $C_{B_{m \times n}}^r$  is reduced by a factor of  $1/4$ . In our proposed binary residual block, the parallel feature extraction first allows the network to capture a more global feature distribution in the horizon and vertical. Then, a  $3 \times 3$  binary convolution is adopted to focus on local regions information. Therefore, we can better encode the spatial information and learn richer feature representation. Besides, it's worth noting that the total complexity of the proposed module is reduced by a factor of  $1/3$ .

**Multi-stage Knowledge Distillation** In this paper, we also propose to increase the representational capability of BSR networks by distilling the information from the full-precision network to the binary network. The teacher network and the student network have a similar architecture, except that the student is low-precision and the teacher is full-

precision. For better illustration, we denote the full-precision network as T and the low-precision network as S. The basic idea is letting S mimic the behavior of T by minimizing the loss between the outputs of T and S. To further constrain the training of S, a multi-stage knowledge distillation strategy is explored, which introduces the loss terms at different stages of the network. The total loss can be expressed as:

$$L_{total} = \lambda L_{CE}(SR, HR) + \gamma \sum_{i=1}^n KL(G_S^i \parallel G_T^i) \quad (4)$$

where  $G_T = \text{softmax}(z_t)$  and  $G_S = \text{softmax}(z_s)$  denote different-level deep features of T and S,  $z_t$  and  $z_s$  are the logits of them. SR denotes the super-resolved image and HR denotes the high-resolution target. We choose the L1 loss as the main loss to measure the difference between SR and HR and use kull-back-Leiber divergence to minimize the distribution loss between T and S.  $\lambda$  and  $\gamma$  are hyper-parameters for balancing the two losses. Besides, considering that the SR task depends on the accuracy of pixel values, we also add a pixel-wise loss between the outputs of student and teacher. In our implementation, the whole function consists of two pixel-based losses and multiple distribution-based losses:

$$L_{total} = \lambda L_{CE}(SR, HR) + \beta L_{CE}(G_S^n, G_T^n) + \gamma \sum_{i=1}^n KL(G_S^i \parallel G_T^i) \quad (5)$$

**Regulable Activation** In order to further make the activation distributions of S more closely match them of T, we introduce a regulable activation to distill information from the teacher network. Given the activation outputs, we readjust them by a regulable activation based on PRelu:

$$y = \begin{cases} x - \theta + \delta & \text{if } x \geq \theta, \\ a(x - \theta) + \delta & \text{otherwise} \end{cases} \quad (6)$$

where  $\theta$  and  $\delta$  are the adaptive variables varying during the training process. By Eq.(6), the network can learn a better distribution depending on the network training (also shown in Fig. 4d). The rectified activations could enhance the model representation ability and thus improve network performance.

## Experiments

### Experimental Setup

**Datasets** DIV2K (Timofte et al. 2017), a high quality image dataset, consists of 800 training images, 100 validation images and 100 test images. Following the setting in (Lim et al. 2017; He et al. 2019), for training, we use the 800 training images from DIV2K. For testing, we employ four benchmark datasets: Set5 (Bevilacqua et al. 2012), Set14 (Zeyde, Elad, and Protter 2012), BSD100 (Martin et al. 2001) and Urban100 (Huang, Singh, and Ahuja 2015).

**Implementation Details** In this paper, to evaluate the effectiveness and universality of the proposed method, we conduct extensive experiments on two types of SR networks, including the interpolation-based method VDSR

Method	Scale	Param	Set5		Set14		B100		Urban100	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Bicubic	$\times 2$	-	33.66	0.930	30.24	0.869	29.56	0.843	26.88	0.840
VDSR	$\times 2$	667k	37.53	0.959	33.05	0.913	31.90	0.896	30.77	0.914
VDSR-BNN*	$\times 2$	667k	36.50	0.952	32.33	0.905	31.25	0.886	29.40	0.894
VDSR-BiReal*	$\times 2$	667k	36.60	0.952	32.41	0.903	31.30	0.886	29.51	0.896
VDSR-BNN+*	$\times 2$	667k	36.81	0.955	32.53	0.907	31.40	0.888	29.62	0.899
VDSR-RTN*	$\times 2$	667k	36.87	0.955	32.59	0.908	31.45	0.889	29.75	0.900
VDSR-IRNet*	$\times 2$	667k	36.94	0.956	32.64	0.909	31.47	0.890	29.80	0.901
VDSR-BTM(ours)	$\times 2$	667k	37.06	0.956	32.72	0.908	31.53	0.889	29.96	0.902
VDSR-IBTM(ours)	$\times 2$	519k	<b>37.24</b>	<b>0.958</b>	<b>32.82</b>	<b>0.911</b>	<b>31.60</b>	<b>0.891</b>	<b>30.14</b>	<b>0.907</b>
Bicubic	$\times 3$	-	30.39	0.868	27.55	0.774	27.21	0.739	24.46	0.735
VDSR	$\times 3$	667k	33.66	0.921	29.77	0.831	28.82	0.798	27.14	0.828
VDSR-BNN*	$\times 3$	667k	32.48	0.906	29.10	0.816	28.21	0.778	25.97	0.793
VDSR-BiReal*	$\times 3$	667k	32.67	0.907	29.25	0.817	28.32	0.779	26.10	0.796
VDSR-BNN+*	$\times 3$	667k	32.84	0.911	29.33	0.822	28.40	0.786	26.24	0.803
VDSR-RTN*	$\times 3$	667k	32.89	0.912	29.41	0.823	28.44	0.786	26.34	0.806
VDSR-IRNet*	$\times 3$	667k	32.88	0.911	29.40	0.822	28.43	0.786	26.30	0.804
VDSR-BTM(ours)	$\times 3$	667k	33.07	0.914	29.53	0.825	28.50	0.788	26.41	0.807
VDSR-IBTM(ours)	$\times 3$	519k	<b>33.23</b>	<b>0.915</b>	<b>29.60</b>	<b>0.827</b>	<b>28.60</b>	<b>0.791</b>	<b>26.64</b>	<b>0.815</b>
Bicubic	$\times 4$	-	28.42	0.810	26.00	0.703	25.96	0.668	23.14	0.658
VDSR	$\times 4$	667k	31.35	0.884	28.01	0.767	27.29	0.725	25.18	0.752
VDSR-BNN*	$\times 4$	667k	30.19	0.858	27.30	0.744	26.70	0.700	24.28	0.715
VDSR-BiReal*	$\times 4$	667k	30.38	0.861	27.41	0.748	26.82	0.705	24.35	0.718
VDSR-BNN+*	$\times 4$	667k	30.42	0.863	27.42	0.750	26.84	0.707	24.37	0.720
VDSR-RTN*	$\times 4$	667k	30.54	0.867	27.53	0.753	26.89	0.710	24.45	0.724
VDSR-IRNet*	$\times 4$	667k	30.66	0.869	27.62	0.757	26.93	0.713	24.56	0.730
VDSR-BTM(ours)	$\times 4$	667k	30.83	0.873	27.76	0.761	27.03	0.717	24.73	0.736
VDSR-IBTM(ours)	$\times 4$	519k	<b>31.06</b>	<b>0.877</b>	<b>27.85</b>	<b>0.762</b>	<b>27.07</b>	<b>0.718</b>	<b>24.88</b>	<b>0.740</b>

Table 1: Quantitative evaluation of VDSR-based state-of-the-art binarization methods. Blod indicates the best performance.

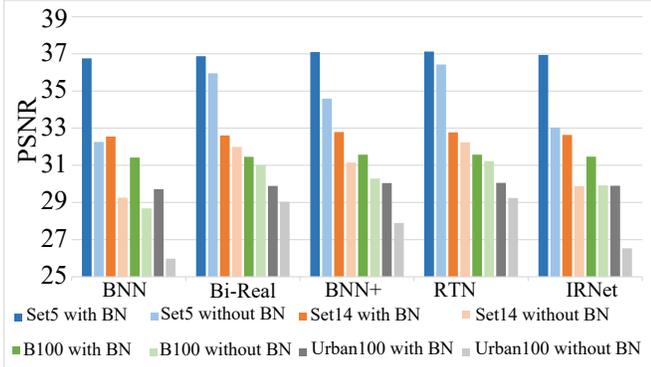


Figure 5: Performance comparison on  $\times 2$  scale datasets of VDSR-based state-of-the-art binarization methods with/without BN

(Kim, Kwon Lee, and Mu Lee 2016b) and the learning-based method EDSR (Lim et al. 2017), which are the most typical networks in SR field. For a fair comparison, we first construct our baseline. For VDSR, we divide the middle 18 convolution layers into 9 baseline blocks and every block contains one short connection. For EDSR, we do not modify the network structure and the residual blocks are directly replaced with our baseline binary blocks. Besides, we adopt the sub-pixel convolution as upscale modules, where the convolution weights are also binary. Then, for our improved binary super-resolution network, the baseline binary blocks

Models	Set5	B100
VDSR	37.06	31.53
VDSR + MRB	37.16	31.57
VDSR + MRB + DL	37.19	31.58
VDSR + MRB + DL + PL	37.21	31.58
VDSR + MRB + DL + PL + RA	37.24	31.60

Table 2: The effect of different components in the proposed method. MRB = Multiple receptive-field block, DL = distribution-based distillation loss, PL = pixle-based distillation loss, RL = regulable activation.

are replaced with our multiple receptive-field block. Besides, the VDSR and EDSR are all divided into four stages for knowledge distillation and the hyper-parameters  $\lambda$ ,  $\beta$  and  $\gamma$  are set to 0.4, 0.6 and 0.005.

**Training Details** In this paper, we reconstruct SR images from LR images with different scaling factors  $\times 2$ ,  $\times 3$  and  $\times 4$ . We obtain LR images by bicubic interpolation and random crop them into a set of  $48 \times 48$ . Due to the training difficulty of large scaling factors, for the interpolation-based method VDSR, the size of  $\times 3$  and  $\times 4$  input patches are  $32 \times 32$  and  $24 \times 24$ . The mini-batch size is set to 16 and the learning rate is set to  $1 \times 10^{-4}$ . For optimization, we use Adam with  $\beta_1 = 0.99$  and  $\beta_2 = 0.999$  and implement all experiments with Pytorch framework on a Telsa V100 GPU.

Method	Scale	Param	Set5		Set14		B100		Urban100	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Bicubic	$\times 2$	-	33.66	0.930	30.24	0.869	29.56	0.843	26.88	0.840
EDSR	$\times 2$	40.12M	38.11	0.960	33.92	0.920	32.32	0.901	32.93	0.935
EDSR-BNN*	$\times 2$	40.12M	34.47	0.938	31.06	0.891	30.27	0.872	27.72	0.864
EDSR-BiReal*	$\times 2$	40.12M	37.13	0.956	32.73	0.909	31.54	0.891	29.94	0.903
EDSR-BNN+*	$\times 2$	40.12M	37.49	0.958	33.00	0.912	31.76	0.893	30.49	0.911
EDSR-RTN*	$\times 2$	40.12M	37.66	0.956	33.13	0.914	31.85	0.895	30.82	0.915
EDSR-BTM(ours)	$\times 2$	40.12M	37.68	0.956	33.20	0.914	31.87	0.895	30.98	0.916
EDSR-IBTM(ours)	$\times 2$	31.73M	<b>37.80</b>	<b>0.960</b>	<b>33.38</b>	<b>0.916</b>	<b>32.04</b>	<b>0.898</b>	<b>31.49</b>	<b>0.922</b>
Bicubic	$\times 3$	-	30.39	0.868	27.55	0.774	27.21	0.739	24.46	0.735
EDSR	$\times 3$	43.07M	34.65	0.928	30.52	0.846	29.25	0.809	28.80	0.865
EDSR-BNN*	$\times 3$	43.07M	20.85	0.399	19.47	0.299	19.23	0.285	18.18	0.307
EDSR-BiReal*	$\times 3$	43.07M	33.17	0.914	29.53	0.826	28.53	0.790	26.46	0.801
EDSR-BNN+*	$\times 3$	43.07M	33.56	0.919	29.73	0.831	28.68	0.794	26.80	0.820
EDSR-RTN*	$\times 3$	43.07M	33.92	0.922	29.95	0.835	28.80	0.797	27.19	0.831
EDSR-BTM(ours)	$\times 3$	43.07M	33.98	0.923	30.04	0.836	28.85	0.798	27.34	0.833
EDSR-IBTM(ours)	$\times 3$	34.68M	<b>34.10</b>	<b>0.924</b>	<b>30.11</b>	<b>0.838</b>	<b>28.93</b>	<b>0.801</b>	<b>27.49</b>	<b>0.839</b>
Bicubic	$\times 4$	-	28.42	0.810	26.00	0.703	25.96	0.668	23.14	0.658
EDSR	$\times 4$	42.48M	32.46	0.897	28.80	0.787	27.71	0.742	26.64	0.803
EDSR-BNN*	$\times 4$	42.48M	17.53	0.188	17.51	0.160	17.15	0.151	16.35	0.163
EDSR-BiReal*	$\times 4$	42.48M	30.81	0.871	27.71	0.760	27.01	0.716	24.66	0.733
EDSR-BNN+*	$\times 4$	42.48M	31.35	0.882	28.07	0.769	27.21	0.724	25.04	0.749
EDSR-RTN*	$\times 4$	42.48M	31.49	0.884	28.14	0.771	27.27	0.726	25.20	0.756
EDSR-BTM(ours)	$\times 4$	42.48M	31.63	0.886	28.25	0.773	27.34	0.728	25.38	0.762
EDSR-IBTM(ours)	$\times 4$	34.09M	<b>31.84</b>	<b>0.890</b>	<b>28.33</b>	<b>0.777</b>	<b>27.42</b>	<b>0.732</b>	<b>25.54</b>	<b>0.769</b>

Table 3: Quantitative evaluation of EDSR-based state-of-the-art binarization methods. Blod indicates the best performance.

Models	PSNR	SSIM	Params
VDSR-BAM	36.60	0.953	668K
VDSR-BTM	37.06	0.956	667K
SRResNet-BAM	37.21	0.956	1547K
VDSR-IBTM	37.24	0.958	519K

Table 4: Comparison between VDSR-BAM and SRResNet-BAM and ours on Set5 test set( $\times 2$ ).

### Comparison with State-of-the-Art Methods

In this paper, we use two typical image quality metrics peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) to evaluate the performance of the super-resolved images. Figure 5 shows the effect of BN layers in existing BNN based SR networks. Table 1 and Table 3 show the comparison between our method and the start-of-the-art binary methods, including BNN (Hubara et al. 2016), Bi-Real (Liu et al. 2018), BNN+ (Darabi et al. 2018), RTN (Li et al. 2020) and IRNet (Qin et al. 2020).

**Comparison with/without BN** In our experiments, we first remove all BN layers to demonstrate the effect of BN layers in existing BNNs. Besides, considering that the Relu will make the input activations only contain +1 values, we adopt the Hardtanh as activation function that is adapted to the sign function. The quantitative comparisons of the performances are summarized in Fig 5. From it we can see when without BN, the existing binary methods suffer serious performance degradation. Among them, Bi-Real is significantly higher than most other methods due to the information protection

capability of shortcuts. RTN obtains the best performance by reparameterizing the weights and activations with scale and offset, which provides a similar effect like BN.

**Comparison on VDSR** In order to apply most existing binary methods into SR networks without BN, we first train all comparison methods by our proposed binary training mechanism. \* denotes the training with our binary training mechanism. Form Table 1 and Figure 5, we can clearly see the performance of all methods is significantly improved by the BTM. Besides, based on the same network, our baseline named VDSR-BTM already surpasses most preciously binarization methods. For our VDSR-IBTM, benefiting from the structure improvement and knowledge distillation, our method could achieve the best on all benchmark datasets over other state-of-the-art BNNs. Besides, we observe that the output activation with our strategies presents a better feature distribution which is similar to it with BN (shown in Fig 4). For scaling factor  $\times 2$ , our VDSR-IBTM achieves 37.24 dB on Set5, which is better than BNN+, RTN and IRNet by 0.43 dB, 0.37 dB and 0.3 dB.

**Comparison on EDSR** During our experiments on EDSR, we find that a complete binary upscale module will lead to a noticeable drop in performance. Therefore, we adopt binary filters and full-precision activations in this paper while the weights and activations are all full-precision in (Xin et al. 2020c). Table 3 shows quantitative comparisons for EDSR. As for our EDSR-IBTM, it can reduce the performance gap with its full-precision counterpart to  $\sim 0.3$  dB on x4 B100, which is powerful enough to become a viable alternative to the full-precision network.

**Comparison with State-of-the-Art BSR** In this paper, we also show a comparison between our method and

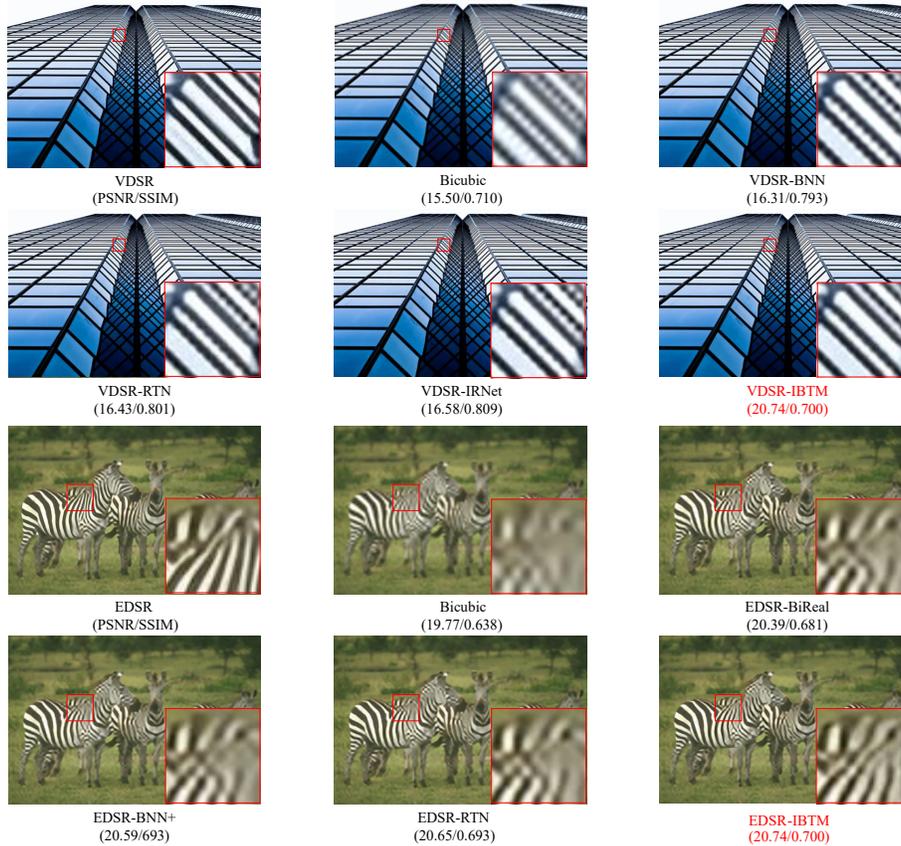


Figure 6: Visual qualitative comparison on  $\times 4$  scale datasets

the start-of-the-art BSR method: bit-accumulation mechanism (BAM) proposed in (Xin et al. 2020c). For VDSR-BAM and VDSR-BTM, our residual block contains one skip-connection while BAM contains multi-skip connections. Compared with VDSR-BAM, our method needs fewer floating-point calculations. From Table 4, our baseline already outperforms it and our VDSR-IBTM outperforms it in a large margin.

**Qualitative Evaluation** In Figure 6, we also show visual comparisons on different testing datasets. It can be found that our method achieves better qualitative results compared with other methods. The SR images using our method are similar to their original full-precision counterparts.

### Ablation Study

In order to study the effects of each part proposed in our work, we gradually modify the model and compare their differences. From Table 3, when we compare the results of the first row and second row, we find the proposed network is efficient in reducing model parameters and improving network performance. It is also observed that benefiting from multi-stage knowledge distillation and regulable activation, the SR networks could obtain excellent performance. Besides, the feature distributions can be adjusted by network training (as shown in Fig 4), which advances the network performance. These all illustrate that our proposed method

is effective for binary super-resolution.

## Conclusion

In this work, we explore a novel training mechanism, helping us train an accurate binary SR network without batch normalization operations. In addition, we construct a strong baseline by combining the highlights of recent binarization methods. We also develop a novel binary network structure and a multi-stage knowledge distillation strategy to reduce the performance gap between the binarized network and the full-precision one.

## Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0103202, in part by the National Natural Science Foundation of China under Grant 61922066, Grant 61876142, Grant 62036007, Grant 61772402, Grant 61671339, and Grant 61976166, in part by the Innovation Capacity Support Plan of Shaanxi Province under Grant 2020KJXX-027 and in part by the Innovation Fund of Xi-dian University.

## References

- Ahn, N.; Kang, B.; and Sohn, K.-A. 2018. Fast, accurate, and lightweight super-resolution with cascading residual network. In *ECCV*.
- Bengio, Yoshua, L. N.; and Courville, A. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.
- Bevilacqua, M.; Roumy, A.; Guillemot, C.; and Alberi-Morel, M. L. 2012. Low-complexity single-image super-resolution based on nonnegative neighbor embedding.
- Chen, T. Q.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. K. 2018. Neural ordinary differential equations. In *NIPS*.
- Darabi, S.; Belbahri, M.; Courbariaux, M.; and Nia, V. P. 2018. Bnn+: Improved binary network training. *arXiv preprint arXiv:1812.11800*.
- Dong, C.; Loy, C. C.; He, K.; and Tang, X. 2015. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*.
- Glorot, X.; and Bengio, Y. 2010. Understanding the difficulty of training deep feedforward neural networks. In *ICLR*.
- Greenspan, H. 2008. Super-resolution in medical imaging. *The Computer Journal*.
- He, X.; Mo, Z.; Wang, P.; Liu, Y.; Yang, M.; and Cheng, J. 2019. Ode-inspired network design for single image super-resolution. In *CVPR*.
- Huang, J. B.; Singh, A.; and Ahuja, N. 2015. Single Image Super-resolution from Transformed Self-Exemplars. In *CVPR*.
- Hubara, I.; Courbariaux, M.; Soudry, D.; El-Yaniv, R.; and Bengio, Y. 2016. Binarized neural networks. In *NIPS*.
- Hui, Z.; Gao, X.; Yang, Y.; and Wang, X. 2019. Lightweight image super-resolution with information multi-distillation network. In *ACM MM*.
- Kim, J.; Kwon Lee, J.; and Mu Lee, K. 2016a. Accurate image super-resolution using very deep convolutional networks. In *CVPR*.
- Kim, J.; Kwon Lee, J.; and Mu Lee, K. 2016b. Deeply-recursive convolutional network for image super-resolution. In *CVPR*.
- Li, Y.; Dong, X.; Zhang, S. Q.; Bai, H.; Chen, Y.; and Wang, W. 2020. RTN: Reparameterized Ternary Network. *AAAI*.
- Lim, B.; Son, S.; Kim, H.; Nah, S.; and Mu Lee, K. 2017. Enhanced deep residual networks for single image super-resolution. In *CVPR*.
- Liu, Z.; Wu, B.; Luo, W.; Yang, X.; Liu, W.; and Cheng, K.-T. 2018. Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. In *ECCV*.
- Martin, D.; Fowlkes, C.; Tal, D.; and Malik, J. 2001. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*.
- Mei, Y.; Fan, Y.; Zhou, Y.; Huang, L.; Huang, T. S.; and Shi, H. 2020. Image super-resolution with cross-scale non-local attention and exhaustive self-exemplars mining. In *CVPR*.
- Qin, H.; Gong, R.; Liu, X.; Wei, Z.; Yu, F.; and Song, J. 2020. IRNet: Forward and Backward Information Retention for Highly Accurate Binary Neural Networks. *CVPR*.
- Rastegari, M.; Ordonez, V.; Redmon, J.; and Farhadi, A. 2016. Xnor-net: Imagenet classification using binary convolutional neural networks. In *ECCV*.
- Rasti, P.; Uiboupin, T.; Escalera, S.; and Anbarjafari, G. 2016. Convolutional neural network super resolution for face recognition in surveillance monitoring. In *International conference on articulated motion and deformable objects*.
- Song, D.; Xu, C.; Jia, X.; Chen, Y.; Xu, C.; and Wang, Y. 2020. Efficient Residual Dense Block Search for Image Super-Resolution. In *AAAI*.
- Tai, Y.; Yang, J.; and Liu, X. 2017. Image super-resolution via deep recursive residual network. In *CVPR*.
- Timofte, R.; Lee, K. M.; Wang, X.; Tian, Y.; Ke, Y.; Zhang, Y.; Wu, S.; Chao, D.; Liang, L.; and Yu, Q. 2017. NTIRE 2017 Challenge on Single Image Super-Resolution: Methods and Results. In *CVPR*.
- Xin, J.; Li, J.; Jiang, X.; Wang, N.; Huang, H.; and Gao, X. 2020a. Wavelet-Based Dual Recursive Network for Image Super-Resolution. *IEEE Transactions on Neural Networks and Learning Systems*.
- Xin, J.; Wang, N.; Jiang, X.; Li, J.; Gao, X.; and Li, Z. 2020b. Facial Attribute Capsules for Noise Face Super Resolution. In *AAAI*, 12476–12483.
- Xin, J.; Wang, N.; Jiang, X.; Li, J.; Huang, H.; and Gao, X. 2020c. Binarized Neural Network for Single Image Super Resolution. In *ECCV*.
- Xu, Z.; and Cheung, R. C. 2019. Accurate and Compact Convolutional Neural Networks with Trained Binarization. *arXiv preprint arXiv:1909.11366*.
- Zeyde, R.; Elad, M.; and Protter, M. 2012. On single image scale-up using sparse-representations. In *Proceedings of the International Conference on Curves and Surfaces*.
- Zhang, J.; Cao, Y.; Zha, Z.-J.; Zheng, Z.; Chen, C. W.; and Wang, Z. 2014. A unified scheme for super-resolution and depth estimation from asymmetric stereoscopic video. *TCVST* 26(3): 479–493.
- Zhang, J.; and Tao, D. 2020. Empowering Things with Intelligence: A Survey of the Progress, Challenges, and Opportunities in Artificial Intelligence of Things. *IEEE Internet of Things Journal* doi:10.1109/JIOT.2020.3039359.
- Zhang, Y.; Li, K.; Li, K.; Wang, L.; Zhong, B.; and Fu, Y. 2018a. Image super-resolution using very deep residual channel attention networks. In *ECCV*.
- Zhang, Y.; Tian, Y.; Kong, Y.; Zhong, B.; and Fu, Y. 2018b. Residual dense network for image super-resolution. In *CVPR*.