
Efficient Image Super Resolution Using Attention In Binary Neural Networks

A Preprint

Zharikov Ilya
Moscow Institute of Physics and Technology

Kirill Ovcharenko
Moscow Institute of Physics and Technology

Abstract

Image Super Resolution [SR] is a crucial class of image processing techniques that enhance quality of visual data. Deep Convolutional Neural Networks [DCNN] have recently shown great results in this field. However, application of DCNN on resource-limited devices remains challenging, as they demand significant amounts of memory, energy and computations. Binary neural networks [BNN] provide a promising approach to reduce computational complexity and speed up the inference of a model. To our best knowledge, there are not many papers devoted to applying BNN to SR tasks, as SR models are much more vulnerable to degradation in performance when decreasing the precision of weights, than image classification models. The paper proposes modification of a convolutional block to make it binary without suffering severe performance decrease.

Keywords Binary Neural Network · Single Image Super Resolution · Binarization · Model compression

1 Introduction

Image Super Resolution [SR] [13] aims to restore High Resolution [HR] image from corrupted Low Resolution [LR] counterpart. This task is important because of its various applications in medical imaging [4] and surveillance instruments [1]. In spite of the research in this field being active, it has not progressed a lot, as some challenges were encountered. Main obstacle is desired output in SR tasks being much more diverse than input, so the model is required to do dense pixel-level prediction, hence is bound to be more complex.

Recent advances in the field of SR owe their success to Deep Convolutional Neural Networks [DCNN] which show state-of-the-art results in a wide range of computer vision problems, such as image classification, Semantic Segmentation etc. However, the models solving these tasks are usually complicated and demand a lot of space and computational resources, thus hindering their implementation on mobile devices, drones and other machines which are limited in GPU memory.

Lately, different methods of reducing complexity of these models were proposed. While some papers focus on pruning and knowledge distillation, other researches introduce quantization as a way to decrease memory needed.

The most extreme form of quantization is binarization. Binary Neural Networks [BNN] use only 1 bit to represent each parameter, thus drastically decreasing space demanded to store the model. Moreover, with all parameters of the model set to $\{-1, 1\}$, most of the calculations can be conducted using XNOR and Bitcount operations. This approach seems promising, as it proposes new ways to design hardware that can help to handle and exploit complex neural networks.

However, it is obvious that BNN sacrifice precision and quality, as they have much less capacity and representational potential than Full-Precision [FP] networks. Previous works in this field propose different methods of maintaining competitive accuracy while achieving better performance. The paper [10] focuses on residual block binarization, which helps to reduce a significant part of the model's parameters. However, full-precision activations keep computational complexity of the model pretty high. ReactNet [9] suggests

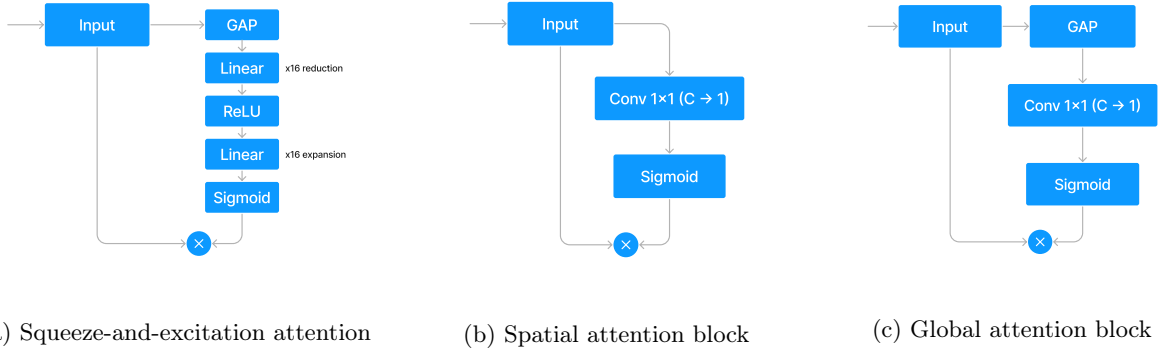


Fig. 1. Attention blocks

generalized binarization and activations functions that help to shift distribution, which significantly increases representational capacity of the binary model. The BBCU [15] proposed effective Convolutional Unit that can be used in any architecture that relies on residual connections. It provides much more efficient training and inference, but oversimplifies weight binarization. Moreover, the block modification implies disposal of the batch normalization block. However, recent studies [14] show that models without the BatchNorm module (e.g. EDSR [8]) suffer a performance drop during binarization as they have diverse distributions between channels and layers. IR-Net [12] reduces information loss by balancing weights to achieve maximum information entropy in forward propagation. After that, IR2Net [16] proposes two essential components of the learning process: Information Restriction and Information Recovery. BNext [5] also applies attention mechanism to obtain the key information from the full-precision activations and smooth out the loss landscape. However, last two papers investigate only the impact of these methods on performance of Image Classification models. Another way of extracting necessary information was proposed in [6], where a squeeze-and-excitation block is added to every transformation to a feature map, so that it can learn dependencies between channels (which are expected to concentrate on different features). In contrast to regular approaches, the paper [18] presented a new method of attention that helps model to better get pixel-level dependencies and exhibits great results in SR task.

This paper adopts some techniques from the researches, mentioned above, and suggests further modifications of convolutional block that help to improve BNN's performance in SR tasks.

- (1) We adopt the idea of using generalized activation function from [9]. That helps to achieve better performance while preserving reasonable computational demands.
- (2) Batch Normalization block was proven to decrease quality of restored images [8]. However, spreading the distribution of binary block's output is necessary for more uniform information flow, as shown in [15]. Thus, we apply scaling block to the output of binary convolution.
- (3) Finally, we advance the idea from [16] and [5] to restrict information from the input to increase learning productivity by implementing attention modules into our binary network. We try different ways to compute attention maps 1, including the methods proposed in [18] and [6].

2 Problem statement

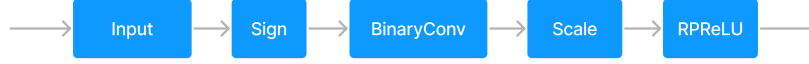
Let $\{(X_i, Y_i)\}_{i=1}^n$ be our image dataset, where $X_i \in \mathbb{R}^{h_i \times w_i \times 3}$ denotes the low resolution image and $Y_i \in \mathbb{R}^{H_i \times W_i \times 3}$ is the high resolution one. Considering M to be the model, SR task targets optimization of

$$Q(M) = \frac{1}{n} \sum_{i=1}^n f(\mathbf{M}(X_i), Y_i) \quad (1)$$

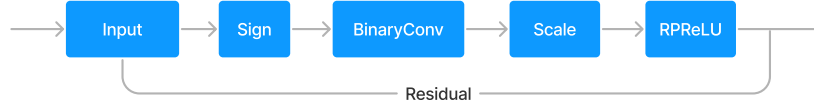
where f represents either PSNR or SSIM metric, defined as:

$$\text{PSNR}(x, y) = 10 \log_{10} \left(\frac{\text{MAX}_I^2}{\text{MSE}(x, y)} \right) \quad (2)$$

Here MAX_I is the maximum valid value for pixel, MSE is mean squared error.



(a) Base block



(b) Base residual block

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (3)$$

μ_x denotes the mean for x , μ_y is the mean for y , σ_x is the variance for x , σ_y is the variance for y , σ_{xy} is the covariation of x and y , c_1 and c_2 - two constants depending on dynamic pixel range.

Now let $B \in \mathcal{B}$ be our binarized representation of model M . \mathcal{B} is space of BNN. Assuming L to be number of layers in $B \in \mathcal{B}$, $W_l \in \{-1, 1\}^{C_{out} \times C_{in} \times K_h \times K_w}$, $l \in \{1, \dots, L\}$. Here C_{out} is the number of output channels, C_{in} is the number of input channels, K_h is the kernel height, K_w is the kernel weight.

Thus, the problem of binarization can be expressed in finding B^* as

$$B^* = \arg \min_{B \in \mathcal{B}} [Q(M) - Q(B)] \quad (4)$$

3 Basic Binary Convolutional Block modification

3.1 Baseline block

In this section we define basic binarization operations that are used to build the Binary Convolutional Block.

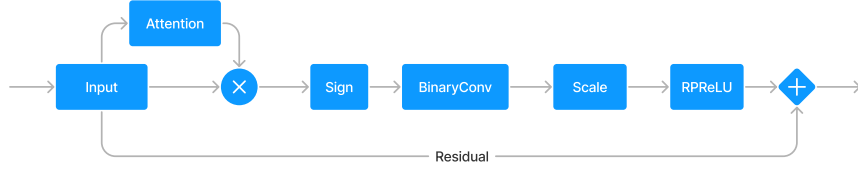
Let $X_t^f \in \mathbb{R}^{H \times W \times C_{in}}$ and $W_t^f \in \mathbb{R}^{K_h \times K_w \times C_{in} \times C_{out}}$ be full-precision activations and full-precision convolution weights on the t -th layer respectively. Here H and W denote the input feature map height and width, C_{in} stands for the number of input channels and C_{out} is the number of output channels. Then $X_t^b \in \{-1, 1\}^{H \times W \times C_{in}}$, $W_t^b \in \{-1, 1\}^{K_h \times K_w \times C_{in} \times C_{out}}$ would be the binary approximations for the corresponding full-precision parameters.

When using binary parameters, convolution operation $X^b * W^b$ can be effectively performed using XNOR and Bitcount operations:

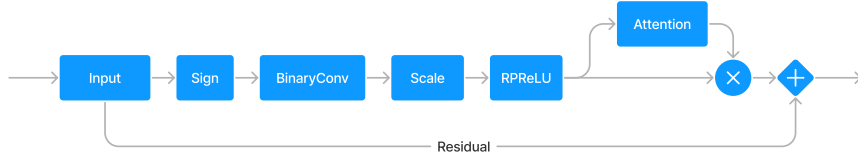
$$X_t^b * W_t^b = \text{Bitcount}(\text{XNOR}(X_t^b, W_t^b))$$

We use regular sign function for activations. Thus, the binary representations can be acquired as follows:

$$x_{i,j,k}^b = \text{Sign}(x_{i,j,k}^f) = \begin{cases} +1, & x_{i,j,k}^f > 0 \\ -1, & x_{i,j,k}^f \leq 0 \end{cases}, \quad i \in [0, H), j \in [0, W), k \in [0, C_{in}) \quad (5)$$



(a) RescalePre block



(b) RescalePost block

Here $x_{i,j,k}^f \in X_t^f$, $x_{i,j,k}^b \in X_t^b$ are single full-precision and binary activations respectively.

Obviously, the derivative of the Sign function cannot be utilized in the training process, as it is impossible to propagate gradients through it. We use an approximation of the Sign derivative, which is defined as follows:

$$\text{Approx} \left(\frac{\partial \text{Sign}(x_{i,j,k})}{\partial x_{i,j,k}} \right) = \begin{cases} 2 + 2x_{i,j,k}, & -1 \leq x_{i,j,k}^f < 0 \\ 2 - 2x_{i,j,k}, & 0 \leq x_{i,j,k}^f < +1, \\ 0, & \text{otherwise} \end{cases}, \quad i \in [0, H), j \in [0, W), k \in [0, C_{in}) \quad (6)$$

When binarizing the convolution weights we use the scaled Sign function:

$$w_{i,j,k,l}^b = \text{Sign}(w_{i,j,k,l}^f) = \begin{cases} +\alpha_l, & w_{i,j,k,l}^f > 0 \\ -\alpha_l, & w_{i,j,k,l}^f \leq 0 \end{cases}, \quad i \in [0, H), j \in [0, W), k \in [0, C_{in}), l \in [0, C_{out}) \quad (7)$$

Here full-precision and binary weights are denoted as $w_{i,j,k,l}^f \in W_t^f$, $w_{i,j,k,l}^b \in W_t^b$ and $\alpha_l \in \mathbb{R}$ represents the scale factor.

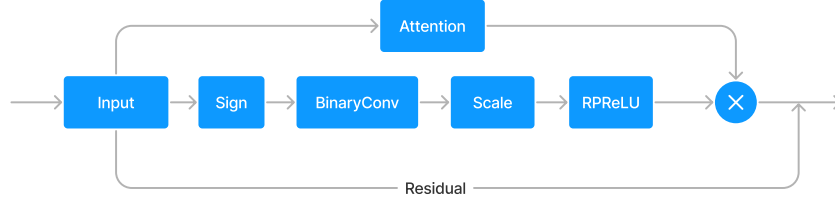
Optimization task of finding the optimal scale factor for binary weights can be expressed as follows:

$$\alpha_l^* = \arg \min_{\alpha} ||W_{t,l}^f - \alpha W_{t,l}^b|| \quad (8)$$

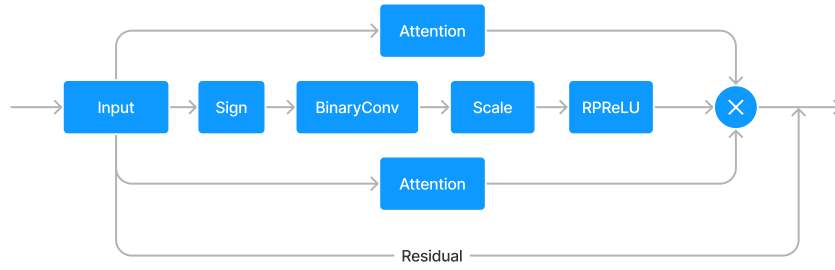
Where $W_{t,l}^f \in \mathbb{R}^{K_h \times K_w \times C_{in}}$, $W_{t,l}^b \in \{-1, 1\}^{K_h \times K_w \times C_{in}}$. We use optimal value $\alpha_l = \alpha^* = \frac{|W_{t,l}^f|}{n}$ to avoid overcomplicating the training process.

We use RPreLU [9] as activation function, because it achieves better performance by shifting the negative component of input's distribution, which is important for BNNs. RPreLU is defined as follows:

$$\text{RPreLU}(x_{i,j,k}) = \begin{cases} x_{i,j,k} - \gamma_k + \zeta_k, & x_{i,j,k} > \gamma_k \\ \beta_k(x_{i,j,k} - \gamma_k) + \zeta_k, & x_{i,j,k} \leq \gamma_k \end{cases}, \quad i \in [0, H), j \in [0, W), k \in [0, C_{in}) \quad (9)$$



(a) Single block



(b) Dual block

Where $x_{i,j,k} \in \mathbb{R}^{H \times W \times C_{in}}$ is an element of the input feature map, $\gamma_k \in \mathbb{R}$ and $\zeta_k \in \mathbb{R}$ are learnable shifts for moving the distribution, and $\beta_k \in \mathbb{R}$ is a learnable coefficient controlling the slope of the negative part.

Previous researches displayed the importance of the residual connection in the Binary Convolution Block, especially in the SR task, so we keep it for every binary convolution to transfer the full-precision information through the block. Moreover, BBCU [15] shows that activation function narrows the negative part of the residual connection, thus losing negative full-precision information. On that account, we keep the idea of moving the residual connection out of the activation function.

EDSR [8] showed that applying Batch Normalization has a negative impact on quality when dealing with pixel-level tasks, such as SR. But the experiments conducted in [15] show that spreading the distribution of values is necessary for BNNs. For that reason, we introduce scale block that applies linear transformation to the output of binary convolution.

Taking in consideration all the points mentioned above, the baseline block can be expressed as follows:

$$X_{t+1}^f = RPreLU(a_t \cdot (X_t^b * W_t^b) + b_t) + X_t^f \quad (10)$$

Here $a_t \in \mathbb{R}$ and $b_t \in \mathbb{R}$ denote learnable Scale block parameters.

3.2 Attention modules

When the full-precision model is being binarized, it is bound to lose some representational capacity and suffer a performance decrease. Previous researches [5, 16] focus on applying attention mechanism to help the model to capture the most important features and dependencies. Further advancing the idea of restricting information, we suggest attention modules that help the model to extract necessary features from the input.

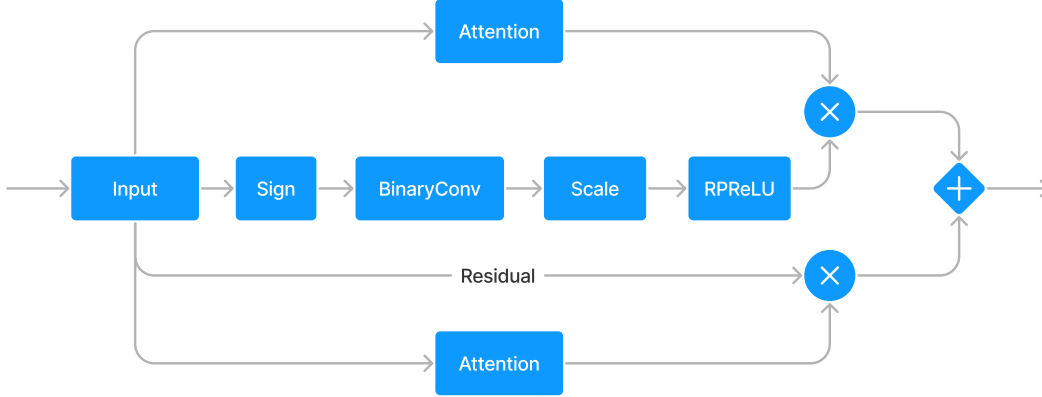


Fig. 5. Both block

We propose several different attention mechanisms: a simple squeeze-and-excitation attention, a spatial attention and a global attention block.

A squeeze-and-excitation block [6] is depicted on 1a. It consists of Global Average Pooling (GAP), two linear layers with a non-linear activation function (ReLU) and a sigmoid function, which is defined as follows:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

The structure of the block helps model to learn non-linear dependencies between channels, hence between different features.

Spatial attention block (1b) consists of one 1×1 convolution with 1 output channel followed by a sigmoid function. It focuses on getting a 2D attention map which helps it to learn relationships between channels for every pixel on the feature map.

Finally, global attention block (1c) has similar structure to the spatial attention block, except for applying Global Average Pooling before a 1×1 convolution. It helps to capture inter-channel dependencies with no regard to the pixel position.

3.3 Proposed modifications

Base and Base-Residual blocks are depicted on 2a and 2b respectively. We propose several modifications that aim to either restrict information from the previous layer or change the distribution of binary convolution inputs and outputs.

Firstly, we investigate the importance of rescaling the inputs and outputs of the binary convolution, proposing RescalePre and RescalePost blocks, that are displayed on 3a and 3b correspondingly.

The outputs of the convolution branch can also contain unnecessary information that should not be propagated to the next layer. Thus, we add attention block that is applied to the RPreLU output before connecting with residual information, getting the modification that is referred to as Single Block 4a.

Dual Block 4b explores the possibility of combining different attention mechanisms to extract various features from the binary information.

In the baseline block, outputs of the binary convolution and activations from the previous layer are added to each other and have the same influence on the result. However, full-precision residual connections have information that cannot be effectively processed by the binary convolution due to its simple structure.

| Modification | GFLOPs | Set5 | | Set14 | | B100 | | Urban100 | |
|----------------------|--------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM |
| EDSR | 45.1 | 38.05 | 0.978 | 33.57 | 0.966 | 32.19 | 0.978 | 32.04 | 0.911 |
| Base | 5.97 | 37.61 | 0.975 | 33.09 | 0.963 | 31.85 | 0.977 | 30.79 | 0.905 |
| Base-Residual | 5.97 | 37.60 | 0.976 | 33.07 | 0.963 | 31.85 | 0.977 | 30.80 | 0.905 |
| RescalePre-Global | 8.11 | 37.55 | 0.976 | 33.03 | 0.963 | 31.82 | 0.977 | 30.74 | 0.904 |
| RescalePre-SE | 8.12 | 37.32 | 0.974 | 32.83 | 0.962 | 31.64 | 0.976 | 30.14 | 0.902 |
| RescalePre-Spatial | 8.14 | 37.58 | 0.976 | 33.04 | 0.963 | 31.80 | 0.977 | 30.68 | 0.905 |
| RescalePost-Global | 8.12 | 37.65 | 0.976 | 33.12 | 0.963 | 31.88 | 0.977 | 30.88 | 0.906 |
| RescalePost-SE | 8.12 | 37.57 | 0.976 | 33.06 | 0.963 | 31.82 | 0.977 | 30.82 | 0.906 |
| RescalePost-Spatial | 8.14 | 37.63 | 0.976 | 33.12 | 0.964 | 31.86 | 0.977 | 30.86 | 0.906 |
| Single-Spatial | 8.14 | 37.64 | 0.977 | 33.11 | 0.964 | 31.87 | 0.977 | 30.84 | 0.906 |
| Single-SE | 8.12 | 37.62 | 0.976 | 33.08 | 0.963 | 31.85 | 0.977 | 30.87 | 0.906 |
| Single-Global | 8.11 | 37.62 | 0.976 | 33.12 | 0.963 | 31.87 | 0.977 | 30.85 | 0.906 |
| Dual-SE-Spatial | 10.29 | 37.61 | 0.976 | 33.09 | 0.963 | 31.84 | 0.977 | 30.83 | 0.906 |
| Dual-Global-Spatial | 10.28 | 37.65 | 0.977 | 33.09 | 0.964 | 31.85 | 0.977 | 30.85 | 0.907 |
| Dual-SE-Global | 10.26 | 37.57 | 0.977 | 33.07 | 0.964 | 31.84 | 0.977 | 30.83 | 0.906 |
| Both-SE-SE | 10.27 | 37.60 | 0.975 | 33.10 | 0.963 | 31.87 | 0.977 | 31.00 | 0.906 |
| Both-SE-Spatial | 10.29 | 37.63 | 0.976 | 33.15 | 0.964 | 31.89 | 0.977 | 31.02 | 0.907 |
| Both-Spatial-SE | 10.29 | 37.68 | 0.976 | 33.17 | 0.964 | 31.91 | 0.978 | 31.08 | 0.908 |
| Both-Spatial-Spatial | 10.31 | 37.68 | 0.976 | 33.18 | 0.965 | 31.88 | 0.977 | 30.93 | 0.906 |
| Both-Global-Global | 10.25 | 37.66 | 0.977 | 33.17 | 0.964 | 31.91 | 0.977 | 31.01 | 0.907 |
| Both-Global-Spatial | 10.28 | 37.62 | 0.977 | 33.13 | 0.964 | 31.89 | 0.977 | 30.90 | 0.906 |
| Both-Spatial-Global | 10.28 | 37.65 | 0.977 | 33.16 | 0.965 | 31.89 | 0.977 | 30.91 | 0.907 |
| Both-Global-SE | 10.26 | 37.62 | 0.975 | 33.13 | 0.964 | 31.88 | 0.977 | 31.02 | 0.908 |
| Both-SE-Global | 10.26 | 37.63 | 0.977 | 33.13 | 0.965 | 31.87 | 0.977 | 31.00 | 0.907 |
| BBCU | 6.0 | 37.67 | 0.976 | 33.14 | 0.964 | 31.89 | 0.977 | 30.92 | 0.906 |

Table 1. Quantitative comparison of PSNR and SSIM on benchmark datasets

Therefore, we need to provide a method to get the most important features from previous layer. Hence, we add an attention module to the residual branch of the convolutional block, obtaining the most advanced modification - Both Block 5.

Having similar structure with regular convolutional block, our modification can be implemented in any SR model architecture with minimal effort. In the next section we apply the modified block to EDSR and prove it to show state-of-the-art results on the benchmark datasets.

4 Computational experiment

4.1 Experiment setup

We use EDSR [8] as a backbone for our modified binary block, as it doesn't require batch normalization module and shows state-of-the-art results in Super-Resolution task. Each convolution of the body is replaced with a modified binary block.

We train our models on DIV2K [2] dataset, which contains 800 images. We evaluate models on 4 benchmark datasets: Set5 [3], Set14 [17], B100 [11] and Urban100 [7].

4.2 Experiment results

Quantitative results are presented in the table 1. The Both-Spatial-SE block shows the best performance among other modifications.

4.3 Ablation study

Quantitative results 2a demonstrate that applying two different attention mechanisms to the output simultaneously (Dual block) performs worse than just applying one mechanism (Single block). The reason behind this is the different purpose of the attention blocks: spatial attention aims to learn inter-channel

| Modification | Urban100 | |
|---------------------|----------|-------|
| | PSNR | SSIM |
| Single-Spatial | 30.84 | 0.906 |
| Single-SE | 30.87 | 0.906 |
| Single-Global | 30.85 | 0.906 |
| Dual-SE-Spatial | 30.83 | 0.906 |
| Dual-Global-Spatial | 30.85 | 0.907 |
| Dual-SE-Global | 30.83 | 0.906 |

(a) Single block VS Dual block

| Modification | Urban100 | |
|---------------------|----------|-------|
| | PSNR | SSIM |
| RescalePre-Global | 30.74 | 0.904 |
| RescalePre-SE | 30.14 | 0.902 |
| RescalePre-Spatial | 30.68 | 0.905 |
| RescalePost-Global | 30.88 | 0.906 |
| RescalePost-SE | 30.82 | 0.906 |
| RescalePost-Spatial | 30.86 | 0.906 |

(b) RescalePre block VS RescalePost block

| Modification | Urban100 | |
|---------------------|----------|-------|
| | PSNR | SSIM |
| RescalePost-Global | 30.88 | 0.906 |
| RescalePost-SE | 30.82 | 0.906 |
| RescalePost-Spatial | 30.86 | 0.906 |
| Single-Spatial | 30.84 | 0.906 |
| Single-SE | 30.87 | 0.906 |
| Single-Global | 30.85 | 0.906 |

(c) RescalePost block VS Single block

Table 2. Quantitative comparison on Urban100 dataset

dependencies for each pixel, while global and SE-attention disregard the spatial position. Combining these blocks can overcomplicate the information and slow down training process.

Comparison of the RescalePre and RescalePost results 2b is consistent with previous researches on the importance of scaling the output of the binary branch. RescalePre block doesn't affect forward pass of the model when using regular Sign function, having influence only on the backward pass. On the other hand, RescalePost block prevents overlapping of the information between two branches, similar to scaling in BBCU [15]. Thus, the RescalePost modification shows better performance by providing more uniform flow of binary and full-precision information.

Contrast between the results of RescalePost and Single blocks 2c displays the importance of applying attention based on the full-precision input. RescalePost block performs rescaling based on the result of the binary block itself, having no direct impact on the input of the block. Yet Single block constructs attention map from the input, thus restricting unnecessary information before propagating further into the block.

Prevalence of the Both block performance shows that both residual and binary branches require information restriction based on the full-precision input.

5 Conclusion

This paper proposes several binary block modifications that improve the performance of Binary Neural Networks in Super Resolution task. We consider attention module being an essential component of the BNN, as it helps to restrict the unnecessary information and provides the network with more effective training process. Quantitative results prove this hypothesis and show that binary neural networks actually suffer from abundance of the input information and perform better when being restricted.

Future researches can be aimed at implementing different attention blocks into the BNN and further experiments on restricting information and learning channel-wise and pixel-wise distribution variations.

References

- [1] A. Aakerberg, K. Nasrollahi, and T. B. Moeslund. Real-world super-resolution of face-images from surveillance cameras. *IET Image Processing*, 16(2):442–452, 2022.
- [2] E. Agustsson and R. Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 126–135, 2017.
- [3] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. 2012.
- [4] F. A. Dharejo, M. Zawish, F. Deebea, Y. Zhou, K. Dev, S. A. Khowaja, and N. M. F. Qureshi. Multimodal-boost: Multimodal medical image super-resolution using multi-attention network with wavelet transform. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2022.
- [5] N. Guo, J. Bethge, C. Meinel, and H. Yang. Join the high accuracy club on imagenet with a binary neural network ticket. *arXiv preprint arXiv:2211.12933*, 2022.
- [6] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.
- [7] J.-B. Huang, A. Singh, and N. Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5197–5206, 2015.
- [8] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017.
- [9] Z. Liu, Z. Shen, M. Savvides, and K.-T. Cheng. Reactnet: Towards precise binary neural network with generalized activation functions. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 143–159. Springer, 2020.
- [10] Y. Ma, H. Xiong, Z. Hu, and L. Ma. Efficient super resolution using binarized neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.
- [11] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001, volume 2*, pages 416–423. IEEE, 2001.
- [12] H. Qin, R. Gong, X. Liu, M. Shen, Z. Wei, F. Yu, and J. Song. Forward and backward information retention for accurate binary neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2250–2259, 2020.
- [13] Z. Wang, J. Chen, and S. C. Hoi. Deep learning for image super-resolution: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3365–3387, 2020.
- [14] R. Wei, S. Zhang, Z. Liu, M. Li, Y. Fan, R. Wang, and R. Huang. Ebsr: Enhanced binary neural network for image super-resolution. *arXiv preprint arXiv:2303.12270*, 2023.
- [15] B. Xia, Y. Zhang, Y. Wang, Y. Tian, W. Yang, R. Timofte, and L. Van Gool. Basic binary convolution unit for binarized image restoration network. *arXiv preprint arXiv:2210.00405*, 2022.
- [16] P. Xue, Y. Lu, J. Chang, X. Wei, and Z. Wei. Ir2net: Information restriction and information recovery for accurate binary neural networks. *arXiv preprint arXiv:2210.02637*, 2022.

- [17] R. Zeyde, M. Elad, and M. Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces: 7th International Conference, Avignon, France, June 24-30, 2010, Revised Selected Papers 7*, pages 711–730. Springer, 2012.
- [18] H. Zhao, X. Kong, J. He, Y. Qiao, and C. Dong. Efficient image super-resolution using pixel attention. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 56–72. Springer, 2020.