Anchor-based Plain Net for Mobile Image Super-Resolution

Zongcai Du Jie Liu Jie Tang* Gangshan Wu
State Key Laboratory for Novel Software Technology, Nanjing Ubiversity, China
{151220022,jieliu}@smail.nju.edu.cn, {tangjie,gswu}@nju.edu.cn

Abstract

Along with the rapid development of real-world applications, higher requirements on the accuracy and efficiency of image super-resolution (SR) are brought forward. Though existing methods have achieved remarkable success, the majority of them demand plenty of computational resources and large amount of RAM, and thus they can not be well applied to mobile device. In this paper, we aim at designing efficient architecture for 8-bit quantization and deploy it on mobile device. First, we conduct an experiment about meta-node latency by decomposing lightweight SR architectures, which determines the portable operations we can utilize. Then, we dig deeper into what kind of architecture is beneficial to 8-bit quantization and propose anchor-based plain net (ABPN). Finally, we adopt quantization-aware training strategy to further boost the performance. Our model can outperform 8-bit quantized FSRCNN by nearly 2dB in terms of PSNR, while satisfying realistic needs at the same time. Code is avaliable at https://github.com/NJU-Jet/SR_Mobile_Quantization.

1. Introduction

Single image super-resolution (SISR) is a classical and long-standing problem in low-level computer vision. The goal is to reconstruct a high-resolution (HR) image according to its degraded low-resolution (LR) counterpart. It has been applied widespreadly in multiple diverse fields, such as HDTV [15], magnetic resonance imaging [49, 24], satellite sensor image reconstruction [10, 30], and underwater applications[6, 29]. The difficulty of SISR is that numerous HR images can map to an identical LR image even through the same degradation model. To find a relatively satisfactory result in the infinite solution space, plenty of traditional SISR algorithms have been proposed in the literature, including but not limited to, interpolation-based [44, 57], image statistics-based [11, 12], patch-based [3, 7, 14] and example-based [13, 4, 41] methods.

Since the dawn of deep learning, CNN-based methods have made further progress in SISR. SRCNN [9] innovatively employed a three-layer CNN to directly learn the mapping function and led to significant improvements compared with conventional methods. After that, more and more creative ideas are introduced, such as residual learning [31, 35, 38], feature fusion [60, 37, 52], well-designed loss function[35, 38, 46] and attention mechanism [58, 8, 19, 55], advancing the performance of SISR.

In recent years, the communities have noticed the deployment issue on mobile device. Most superior models are designed for desktop purposes so they can not be directly applied to mobile environment. In order to strive towards the ultimate goal of applying SISR technology to real-world applications where computational resources are limited, image restoration on smart-phone contests [23, 43, 56] have been held to shed a light upon this problem. Meanwhile, most mobile devices are embedded with deep learning accelerators, and some benchmark suites [22, 45] are developed to measure their performances. For creating mobile-friendly models, there are two basic ideas. One is toward network optimization which can be mainly categorized into quantization [27, 48], pruning [2, 5] and knowledge distillation [42, 54]. The other is toward lightweight architecture design[20, 19, 1, 55]. Our focus is how to create a general SISR network architecture which is beneficial to 8-bit quantization.

In spite of achieving prominent improvements, yet there are drawbacks in two aspects. First, the obtained models are usually evaluated on desktop CPUs and GPUs, making it nearly impossible to estimate the actual inference time and memory consumption on real mobile hardware. Second, even recent state-of-the-art (SOTA) lightweight models include dozens of convolution nodes [38, 1, 20] and time-consuming nodes such as attention [19, 55], making them impracticable for realistic use-cases (*e.g.* restore twenty-four 1080P video frames per second). We resolve the drawbacks by researching meta-node latency on mobile hardware and digging deeper into what kind of architecture can really make sense to INT8 quantization. In summary, our main contributions are as follows:

*Corresponding author

- We investigate meta-node latency on mobile hardware according to SOTA lightweight SISR architectures, which yields portable operations.
- We propose anchor-based residual learning strategy, which is much faster than nearest neighbor resize on mobile device, and can largely improve the INT8 quantized model performance by nearly 2dB without any parameter cost.
- We propose anchor-based plain net (ABPN) for mobile SISR, which is able to restore twenty-seven 1080P images (x3 scale) per second, maintaining good perceptual quality as well.

2. Related Work

2.1. Overview of image super-resolution

Recently, deep learning based methods have achieved dramatic improvements in various kinds of tasks including SISR. Dong *et al*. [9] innovatively introduced a deep learning model called SRCNN to reconstruct HR image in an end-to-end manner. Although SRCNN outperforms handcrafted models by a large margin, it entails high computational loads due to learning in the HR space. To solve the problem, shi *et al*. proposed ESPCN [47] to replace the bicubic filter with a more efficient sub-pixel convolution. In the same period, Kim *et al*. [31] deepened the network to twenty layers, indicating that the depth is crucially important for SISR task. Subsequently, Ledig *et al*. [35] introduced residual block (RB) [17] to maximize the power of residual learning. Furthermore, based on SRResNet [35], Lim *et al*. [38] presented an enhanced deep super-resolution network (EDSR), which made a breakthrough by removing unnecessary modules in RB and had a far-reaching impact on the succeeding studies [60, 58, 1, 59, 39, 55]. For example, RDN [60] proposed residual dense block to make full use of all the hierarchical features via dense connected convolution layers. RCAN [58] integrated channel attention mechanism into RB and adopted residual-in-residual structure to form a very deep network.

2.2. Lightweight image super-resolution

Due to the growing realistic demand, many works have devoted to making models lighter and faster. They can be divided into explicit [32, 51, 37, 1, 20] and implicit schemes [34, 1, 52, 19, 55]. The former adopts simple operations to explicitly reduce the model complexity, such as directly cutting down the width and depth [37], recurrent structure [32, 51] and group convolution [1, 20]. These naive strategies bring about either accuracy loss or more extra overheads (*e.g*., FLOPs). The latter implicit scheme concentrates on sufficiently utilizing intermediate features as well as enhancing the discriminative capability of the network, thus leading to less computational cost and better results on the whole. For instance, LapSRN [34] exploited features at each pyramid level to restore the subband residuals of different high-resolution images. MemNet [52] introduced gating mechanism to bridge the long-term with short-term information. CARN [1] implemented cascading mechanism to incorporate features at both local and global level. IMDN [19] retained partial information as refined features and fused the distilled features by contrast-aware channel attention (CCA) mechanism. LatticeNet [55] created a butterfly structure and also applied CCA to dynamically combine two RBs. The capacity of lightweight models is limited, so recent architecture designs pay attention to making full use of information of different levels as has been stated above. However, situation on mobile hardware is totally different from desktop CPUs and GPUs. For example, hierarchical feature fusion [1, 20, 19, 55] would cause slow access to RAM due to limited cache memory on mobile device. Another popular strategy, attention mechanism [58, 28, 19, 55], would also lead to unbearable time overhead because of calculating global statistics and element-wise multiplication.

2.3. Network Quantization

Quantization is a process of distributing continuous real-valued infinite numbers to a smaller set of discrete finite values, for minimizing the number of bits required and also maximizing the accuracy of the attendant computations. The widely used full-integer quantization technology can be three times faster than original float-point network, and holds the potential to reduce memory footprint by a factor of 4x. Post-training quantization and quantization-aware training are two famous techniques supported by TensorFlow [26]. The former estimates value ranges of network parameter and activation by traversing provided representative data after training, while the latter finish this process during training by inserting fake quantization nodes. Both of them have demonstrated promising results on image perception tasks [18, 25, 50], but applying them to SR task is much harder and will incur significant accuracy drop. The reason is that current architectures remove batch normalization (BN) layers because they result in blurred reconstructed HR images with artifacts [38], but the removal also leads to high dynamic quantization range at the same time. There are limited works [40, 36] targeting on solving this problem. [40] binarizes the convolution filters only in residual blocks, and adopts a learnable weight for each binary filter, which can not be applied to full-integer device. [36] proposes PArameterized Max Scale to explore the upper bound of the quantization range adaptively, which increase training complexity due to manually selecting hyper-parameter and structured knowledge transfer. Simple and useful technology still remains to be explored.

3. Proposed Method

In this section, we start from investigating meta-node latency. Then, we introduce the insight behind our anchor-based residual learning. Finally, we build our final ABPN and describe the design principle of each component.

3.1. Meta-node Latency

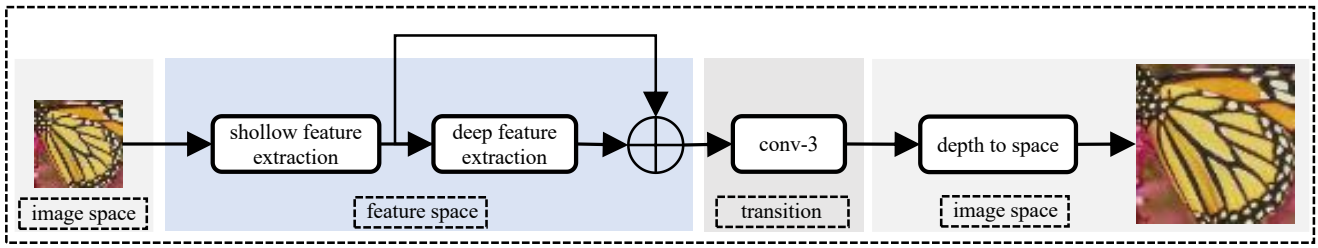
We bear in mind that our goal is to create a real-time model qualified for realistic use-cases (*e.g.* super-resolve video frames). The first thing is figuring out the set of portable meta-node and time-consuming meta-node. We create our initial meta-node set by decomposing recent lightweight SR architectures [38, 1, 20, 19, 55] and then test these meta-nodes on Synaptics Dolphin Platform with a dedicated NPU. They can be divided into four categories: tensor operator nodes, convolution nodes, activation nodes and resize nodes. From Table. 1, we have four observations. First of all, recent technologies used in SOTA lightweight architectures seem to be impracticable to be deployed on mobile device. EDSR [38] adopts a mass of RBs, and each RB will introduce an element-wise addition which is even slower than highly-optimized convolution layer. CARN incorporates global and local features, and each incorporation includes one concatenation of large amount of channels and one 1×1 convolution, bringing about only 0.09dB improvement according to their article. IDN [20] and IMDN [19] are also in dire straits on mobile device, for rapid feature split and concat. It's more serious for LatticeNet [55] which adopts sixteen CA blocks, and each CA block contains one element-wise addition and multiplication, two pooling layers, and four 1×1 convolution layers. Total sixteen CA blocks can only contribute to 0.15dB improvement while leading to heavy computational burden. Another common problem is that they all need to retain features of previous layers, and utilize 1×1 convolution layer to control how much of the previous states should be reserved, and determine how much of the current state should be stored. This long-term dependency causes frequent slow contaction with RAM since there is only limited cache memory in mobile device. Thus, we would not take feature fusion, feature distillation, group convolution and attention mechanism into consideration. Second, although the number of parameters and floating point operations of 3×3 convolution layer is nine times as large as that of 1×1 convolution layer, the time consumption does not differ much due to parallel calculation. So we prefer to utilize 3×3 convolution layer to produce larger receptive fields, which is critically important for micro-architecture. Third, as for activation function, we choose ReLU because it is much faster than Leaky ReLU and we find that the performance gain of Leaky ReLU is quite small (within 0.03dB). Last, resize nodes are too slow because of coordinate mapping between interpolated HR image and input LR image.

Table 1. Meta-nodes inference time (ms) on Synaptics Dolphin Platform. Resize nodes are applied to network input, while other nodes are applied to 1080P network output.

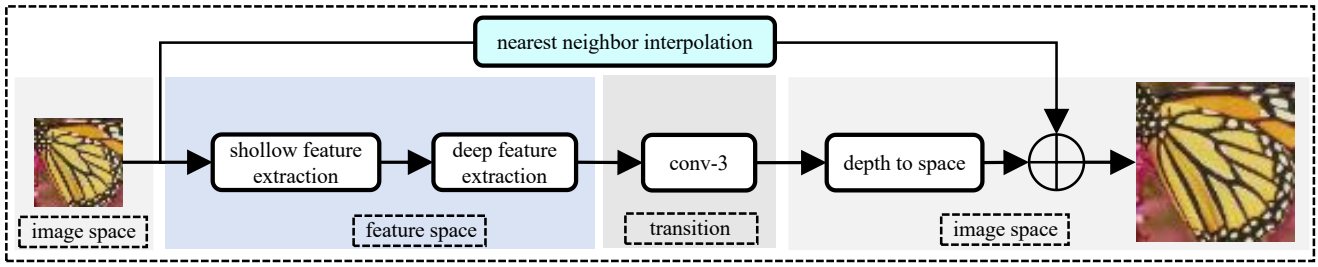
Main type	Meta-node	Time
Tensor operator nodes	Channel split	9.8
	Channel concat	10.4
	Add two tensors	5.2
	Multiply two tensors	9.6
	Global max pooling	20.0
	Global average plooling	13.1
Convolution nodes	3×3 Convolution	4.3
	1×1 Convolution	2.9
Activation nodes	ReLU	1.3
	Leaky ReLU	3.6
Resize nodes	Nearest neighbor	57.6
	Bilinear	75.4

3.2. Anchor-based Residual Learning

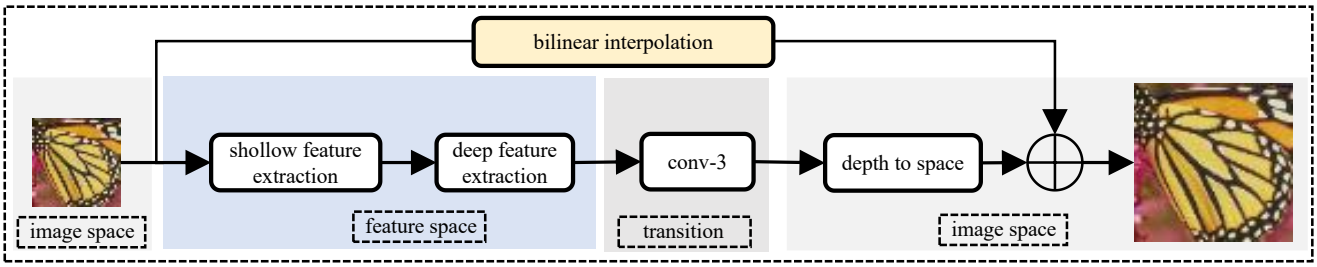
As has been discussed in Sec. 3.1, the available meta-nodes are really limited. To get a good solution, we need to dig deeper into the relationship between architecture design and INT8 quantization. As we know, the difficulty lies in high dynamic range of image-to-image mapping, so the direct idea is to produce lower standard deviation weights and activations. There are two simple ways to achieve this goal. One is adding BN layer, and the other is residual learning. On the one hand, BN is always integrated into RB, so the introduction will not only induce extra time and memory overhead, but also significantly decrease the performance by about 0.2dB. On the other hand, neighboring pixels always have nearly the same values so it seems nature to learn residual, which is close to zero. Residual learning can be divided into image-space residual learning (ISRL) and feature-space residual learning (FSRL). ISRL is adopted in early works [31, 51] to map a LR image to a blurred HR image, while FSRL is widely adopted in recent SOTA models [35, 20, 19] which sightly outperforms ISRL in floating-point space. However, we argue that ISRL is better for INT8 quantization because it forces the whole network to learn small residual, and this intuition will be experimentally verified in Sec. 4. From Table 1, we can see that both image space interpolations suffer from unbearable time cost and even just one single node can not satisfy realistic demand. We recognize it is the floating calculation in coordinate mapping that restricts the deployment of ISRL. To tackle this problem, we propose anchor-based residual learning (ABRL). Different from nearest neighbor interpolation which needs floating calculation in coordinate mapping, ABRL directly repeats every pixel nine times in LR space to generate anchors for every pixel in HR space. Thanks to unique pixel shuffle layer, our ABRL can be easily realized by one channel-concat and one addition meta-



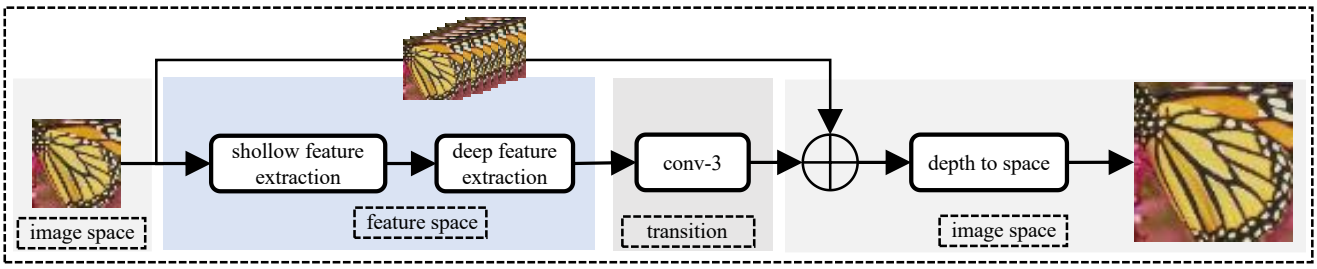
(a) Residual learning in feature space



(b) Nearest neighbor interpolation in image space



(c) Bilinear interpolation in image space



(d) Proposed anchor-based residual learning in image space

Figure 1. Illustration of residual learning in image space and feature space.

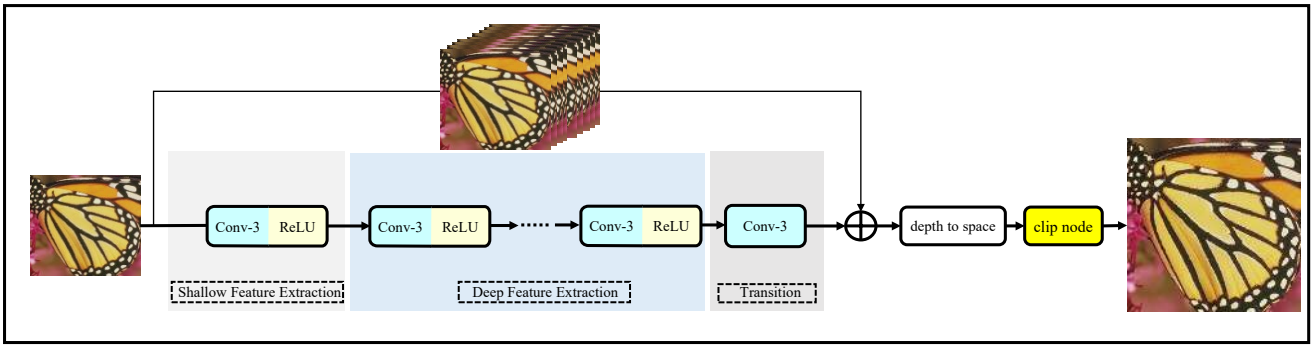


Figure 2. The whole network architecture.

node. Fig. 1 shows the four different kinds of residual learning strategy. As for time overhead, residual learning in feature space contains only one element-wise addition, which costs 5.2ms. Our proposed ABRL contains one channel-concat and one element-wise addition, which total costs 15.6ms, nearly a quarter of the time cost of nearest neighbor meta-node. It should be noted that the cost of nine channel-concats in LR image space is almost equal to that of one channel-concat in HR image space. Our ABRL has two main advantages: the first is that it can largely improve the performance of INT8 quantized model compared with that of residual learning in feature space (up to 0.6dB); the second is that multi-branch architecture can be inferred parallelly so the actual cost of ABRL is the same as feature-space residual learning, and the main cost of our ABRL and FSRL is brought by slow access to RAM. It is also worth mentioning that our ABRL is a special case of nearest neighbor interpolation when the scale factor is integer.

3.3. Network Architecture

The whole architecture is depicted in Fig. 2. Our architecture mainly consists of four parts: shallow feature extraction part which transfers the image to feature space, deep feature extraction which extracts high-level information and restore details (edges, textures) step by step, reconstruction part which maps features to HR image space, and post processing part which re-arrange pixels and restricts the values within normal image range. Let's denote I_{LR} and I_{SR} as the input and output of our ABPN. We obtain shallow feature F_0 by:

$$F_0 = H_{SFE}(I_{LR}) \tag{1}$$

where $H_{SFE}(\cdot)$ denotes the mapping function in shallow feature extraction. We use one 3×3 convolution layer followed by ReLU to form this part. After that, we use Conv-ReLU pairs, which is the fastest combination in Table. 1, to gradually refine details. The i-th deep features F_i is obtained through:

$$F_i = H_{DFE_i}(F_{i-1}), i = 1, \ldots, 5, \tag{2}$$

where $H_{DFE_i}(\cdot)$ represents i-th Conv-ReLU pair in deep feature extraction part.To fully take advantage of parallel inferring, we set the number of Conv-ReLU pairs to 5 to match the overhead in the upper branch, which means when Conv-ReLU pairs is less 5, the mobile inference time remains the same. Then, one convolution layer is adopted to transfer features to HR image space:

$$F_t = H_T(F_5) \tag{3}$$

where $H_T(\cdot)$ is the mapping function in transition layer and F_t is the obtained residual image features.Our ABRL is applied subsequently to get the super-resolved image of which the spatial pixels are put in the channel axis:

$$F_{SR} = F_t + I_{LR} \tag{4}$$

Finally, pixel shuffle layer is used to re-arrange F_{SR} and a clip node is used to restrict values to get I_{SR}:

$$I_{SR} = H_{PP}(F_{SR}) \tag{5}$$

where $H_{PP}(\cdot)$ is the mapping function in post processing part. Clip node, at the tail of the network,clips values less than zero or larger than 255. The absence of this node will cause the shift of output distribution, and when applying full-integer quantization the converter would think that there are negative values of real images.

3.4. Loss Function

There are a lot of loss functions which have been adopted in previous works [38, 20, 19]. To make sure the improvement is mainly from our design, we simply use L_1 loss function to optimize our network which can be formulated as:

$$L(\Theta) = \frac{1}{N} \sum_{i=1}^{N} \left\| f_{ABPN}(I_{LR}^i) - I_{HR}^i \right\|_1 \tag{6}$$

where Θ denotes the parameters of our network and N is the total number of training samples. I_{LR}^i and I_{HR}^i denote the i-th LR patch and the corresponding ground truth. $f_{ABPN}(\cdot)$ represents the operations of the proposed ABPN.

4. Experiments

4.1. Settings

Implementation details. In each training batch, 16 cropped 64×64 LR RGB patches augmented by random flipping and rotation are sent to the network. The learning rate is initialized as 1×10^{-3} and decreases half per 200 epochs for 1000 epochs. The number of kernels in the residual learning branch is set to 28. Parameters of our model is initialized using the method proposed by He *et al*. [16] and optimized by ADAM optimizer [33] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. We follow the Mobile AI image super-resolution challenge [21] to measure SR results in the RGB space, and adopt DIV2K [53] as training and validation sets.

4.2. Residual Learning

In this section, we verify the efficiency of residual learning and the superiority of our ABRL. We first remove the upper branch in Fig. 2 to build our baseline model. Then, we separately add four residual learning strategies to the baseline model. The results are reported in Table. 2, from which we have the following observations. For FP32 model, FSRL model can achieve the best performance (+0.03dB), while other methods achieves nearly the same performance. For INT8 quantization model, architecture without residual learning suffer from severly accuracy drop (-1.93dB), while the architectures with feature-space residual learning drops 0.78dB and architectures with image-space residual learning drop only 0.13dB. Thus, we can conclude that residual learning can largely allevate high dynamic range problem in INT8 quantization, and image-space residual learning is much better than feature-space residual learning.

Table 2. Investigation of image space and feature space residual learning. The inference time is measured by Synaptics smart TV platform. FSRL denotes feature-space residual learning in Fig. 1, ABRL denotes our proposed anchor-based residual learning.

Model	Params	FP32	INT8	Inference time
Baseline	42.54K	30.21	28.28	26ms
Baseline+nearest	42.54K	30.22	30.09	57.6ms
Baseline+bilinear	42.54K	30.24	30.11	74.9ms
Baseline+FSRL	42.54K	30.27	29.49	35.9ms
Baseline+ABRL	42.54K	30.22	30.09	36.8ms

4.3. Quantize-Aware Training

Quantization-aware training (QAT) is a popular technology to boost the performance without any inference stage cost. We set initial learning rate to 1×10^{-4} and decreases half per 50 epochs for 200 epochs. We can further improve the performance by 0.06dB. By now, the INT8 quantized network only lose 0.07dB on mobile image super-resolution compared with its floating point version.

4.4. Test on Snapdragon 820

We report the inference time of our model on a real mobile device with Snapdragon 820. We use AI Benchmark[1] to get the CPU, GPU, NNAPI running time. The results are shown in Table. 3.

Table 3. Inference time (ms) of ABPN on Snapdragon 820.

Model	CPU	GPU	NNAPI	PSNR
FSRCNN	149.2	44.3	21.7	28.1
ABPN	235.0	69.8	39.2	30.15

4.5. MAI2021 SISR Challenge

This work is initially proposed for the purpose of participating in the MAI2021 Single Image Super-Resolution Challenge [21]. We report the preliminary results and our result in Table. 4. Our first submission is the same model without the clip node at the tail of the network, and the results is really bad (less than 20dB). We solve this issue after the deadline and send the corrected model to the organizers. Beneficial from our anchor-based residual learning, our models can outperform other models by a large margin especially for PSNR metric. Also, we can achieve the fastest inference speed.

Table 4. Comparison of our results and official preliminary results on MAI2021 Single Image Super-Resolution track.

	PSNR	SSIM	NPU Runtime	Score
ABPN (**Ours**)	29.87	0.8686	36.89	92.72
deepernewbie	29.58	0.86	44.85	51.02
JeremieG	29.41	0.8537	38.32	47.18
richlaji	29.52	0.8607	62.25	33.82
xindongzhang	28.82	0.8428	76.61	10.41

4.6. Visual Comparision

We show visual comparison of our int8 quantized model and int8 quantized FSRCNN in Fig. 3. Our methods can reconstruct more textures and faithfully edges, which demonstrates the superiority of our proposed ABPN.

5. Conclusion

We propose an efficient network called anchor-based plain net (ABPN) for INT8 quantization. The key component is anchor-based residual learning (ABRL), which realize the same functionality of image-space residual learning while being as fast as feature-space residual learning. Our INT8 quantization network can achieve nearly the same performance as original floating-point network.

[1] https://ai-benchmark.com/download

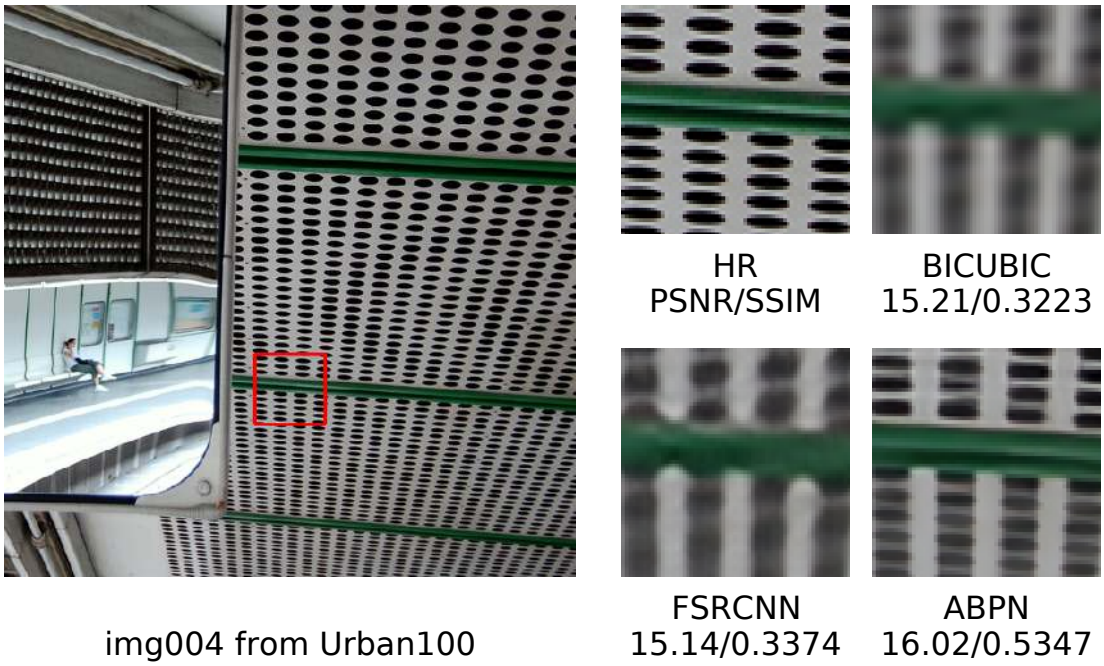


Figure 3. Visual results on Urban100 of int8 model.

References

[1] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *ECCV (10)*, volume 11214 of *Lecture Notes in Computer Science*, pages 256–272. Springer, 2018. 1, 2, 3

[2] Y. Bhalgat, Y. Zhang, J. Lin, and F. Porikli. Structured convolutions for efficient neural network design. 2020. 1

[3] Hong Chang, Dit Yan Yeung, and Yimin Xiong. Super-resolution through neighbor embedding. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004. 1

[4] Hong Chang, Dit Yan Yeung, and Yimin Xiong. Super-resolution through neighbor embedding. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004. 1

[5] S. K. Chao, Z. Wang, Y. Xing, and G. Cheng. Directional pruning of deep neural networks. 2020. 1

[6] Yuzhang Chen, Wei Li, Min Xia, Qing Li, and Kecheng Yang. Super-resolution reconstruction for underwater imaging. *Optica Applicata*, 41(4):841–853, 2011. 1

[7] D. Dai, R. Timofte, and L. Van Gool. Jointly optimized regressors for image super-resolution. *Computer Graphics Forum*, 34(2):95–104, 2015. 1

[8] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[9] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *ECCV (4)*, volume 8692 of *Lecture Notes in Computer Science*, pages 184–199. Springer, 2014. 1, 2

[10] Hala M. Ebied, Ashraf Helmy, Taymoor M. Nazamy, Mohamed Fahmy Tolba, and Marwa Sayed. Satellite super resolution image reconstruction based on parallel support vector regression. In *International Conference*, 2014. 1

[11] N. Efrat, D. Glasner, A. Apartsin, B. Nadler, and A. Levin. Accurate blur models vs. image priors in single image super-resolution. In *2013 IEEE International Conference on Computer Vision (ICCV)*, pages 2832–2839, Los Alamitos, CA, USA, dec 2013. IEEE Computer Society. 1

[12] Carlos Fernandezgranda and Emmanuel Candès. Super-resolution via transform-invariant group-sparse regularization. 2013. 1

[13] W. T Freeman, T. R Jones, and E. C Pasztor. Example-based super-resolution. *Computer Graphics and Applications IEEE*, 22(2):56–65, 2002. 1

[14] X. Gao. Image super-resolution with sparse neighbor embedding. *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society*, 21(7):3194, 2012. 1

[15] Tomio Goto, Takafumi Fukuoka, Fumiya Nagashima, Satoshi Hirano, and Masaru Sakurai. Super-resolution system for 4k-hdtv. In *International Conference on Pattern Recognition*, 2014. 1

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. 2015. 6

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 2

[18] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. 2017. 2

[19] Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang. Lightweight image super-resolution with information multi-distillation network. In *ACM Multimedia*, pages 2024–2032. ACM, 2019. 1, 2, 3, 5

[20] Zheng Hui, Xiumei Wang, and Xinbo Gao. Fast and accurate single image super-resolution via information distillation network. In *CVPR*, pages 723–731. IEEE Computer Society, 2018. 1, 2, 3, 5

[21] Andrey Ignatov, Radu Timofte, Maurizio Denna, and Abdel Younes. Real-time quantized image super-resolution on mobile npus, mobile ai 2021 challenge: Report. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2021. 6

[22] A. Ignatov, R. Timofte, A. Kulik, S. Yang, K. Wang, F. Baum, M. Wu, L. Xu, and L. Van Gool. Ai benchmark: All about deep learning on smartphones in 2019. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3617–3635, 2019. 1

[23] A. Ignatov, R. Timofte, T. V. Vu, T. M. Luu, and C. Jung. Pirm challenge on perceptual image enhancement on smartphones: Report. *Springer, Cham*, 2018. 1

[24] Rafiqul Islam, Andrew J Lambert, Mark R Pickering, Jennie M Scarvell, and Paul N Smith. A wavelet-based super-resolution method for multi-slice mri. *Journal of Biomedical ence and Engineering*, 5(5(12A)):862–870, 2012. 1

[25] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. 2017. 2

[26] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew G. Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. *CoRR*, abs/1712.05877, 2017. 2

[27] K. Jia and M. Rinard. Efficient exact verification of binarized neural networks. 2020. 1

[28] Jie, Hu, Li, Shen, Samuel, Albanie, Gang, Sun, Enhua, and Wu. Squeeze-and-excitation networks. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 2

[29] Wenjing Kang and Gongliang Liu. Super-resolution information collection in underwater sensor networks with random node deployment: A compressed sensing approach. *Journal of Networks*, 7(8), 2012. 1

[30] Hatem Magdy Keshk and Xu Cheng Yin. Satellite super-resolution images depending on deep learning methods: A comparative study. In *2017 IEEE International Conference*

on Signal Processing, Communications and Computing (ICSPCC), 2017. 1

[31] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, pages 1646–1654. IEEE Computer Society, 2016. 1, 2, 3

[32] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *CVPR*, pages 1637–1645. IEEE Computer Society, 2016. 2

[33] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *Computer ence*, 2014. 6

[34] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *CVPR*, pages 5835–5843. IEEE Computer Society, 2017. 2

[35] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, and Zehan and Wang. Photo-realistic single image super-resolution using a generative adversarial network. 2017. 1, 2, 3

[36] H. Li, C. Yan, S. Lin, X. Zheng, B. Zhang, F. Yang, and R. Ji. Pams: Quantized super-resolution via parameterized max scale. In *Springer, Cham*, 2020. 2

[37] Zhen Li, Jinglei Yang, Zheng Liu, Xiaomin Yang, Gwanggil Jeon, and Wei Wu. Feedback network for image super-resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2

[38] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. 2017. 1, 2, 3, 5

[39] Jie Liu, Wenjie Zhang, Yuting Tang, Jie Tang, and Gangshan Wu. Residual feature aggregation network for image super-resolution. pages 2356–2365, 06 2020. 2

[40] Y. Ma, H. Xiong, Z. Hu, and L. Ma. Efficient super resolution using binarized neural network. 2018. 2

[41] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. Non-local sparse models for image restoration. In *2009 IEEE 12th International Conference on Computer Vision (ICCV)*, 2010. 1

[42] H. Mobahi, M. Farajtabar, and P. L. Bartlett. Self-distillation amplifies regularization in hilbert space. 2020. 1

[43] S. Nah, S. Son, R. Timofte, K. M. Lee, and T. Huck. Ntire 2020 challenge on image and video deblurring. *IEEE*, 2020. 1

[44] Deepu Rajan and Subhasis Chaudhuri. Generalized interpolation and its application in super-resolution imaging. *Image and Vision Computing*, 19(13):957–969, 2001. 1

[45] V. J. Reddi, C. Cheng, D. Kanter, P. Mattson, and G. Schmuelling. Mlperf inference benchmark. In *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*, pages 446–459, 2020. 1

[46] Mehdi S. M. Sajjadi, Bernhard Scholkopf, and Michael Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. In *IEEE International Conference on Computer Vision*, 2017. 1

[47] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. 2016. 2

[48] M. Shkolnik, B. Chmiel, R. Banner, G. Shomron, Y. Nahshan, A. Bronstein, and U. Weiser. Robust quantization: One model to rule them all. 2020. 1

[49] Andre Souza and Robert Senn. Model-based super-resolution for mri. *Conference proceedings: ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, 2008:430–434, 2008. 1

[50] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *IEEE*, pages 2818–2826, 2016. 2

[51] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *CVPR*, pages 2790–2798. IEEE Computer Society, 2017. 2, 3

[52] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *ICCV*, pages 4549–4557. IEEE Computer Society, 2017. 1, 2

[53] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming Hsuan Yang, and Qi Guo. Ntire 2017 challenge on single image super-resolution: Methods and results. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017. 6

[54] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. 2020. 1

[55] Luo Xiaotong, Xie Yuan, Zhang Yulun, Qu Yanyun, Li Cuihua, and Fu Yun. Latticenet: Towards lightweight image super-resolution with lattice block. 2020. 1, 2, 3

[56] Kai Zhang and Nan Nan. AIM 2019 challenge on constrained super-resolution: Methods and results. In *2019 IEEE/CVF International Conference on Computer Vision Workshops, ICCV Workshops 2019, Seoul, Korea (South), October 27-28, 2019*, pages 3565–3574. IEEE, 2019. 1

[57] Lei Zhang and Xiaolin Wu. An edge-guided image interpolation algorithm via directional filtering and data fusion. *IEEE Trans Image Process*, 15(8):2226–2238, 2006. 1

[58] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. 2018. 1, 2

[59] Yulun Zhang, Kunpeng Li, Kai Li, Bineng Zhong, and Yun Fu. Residual non-local attention networks for image restoration. 2019. 2

[60] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. 2018. 1, 2